GigaScience

Iterative Hard Thresholding in GWAS: Generalized Linear Models, Prior Weights, and Double Sparsity

--Manuscript Draft--

Manuscript Number:	GIGA-D-19-00398
Full Title:	Iterative Hard Thresholding in GWAS: Generalized Linear Models, Prior Weights, and Double Sparsity
Article Type:	Technical Note
Funding Information:	
Abstract:	p.p1 {margin: 0.0px 0.0px 0.0px 0.0px; font: 8.0px Helvetica} Background: Consecutive testing of single nucleotide polymorphisms (SNPs) is usually employed to identify geneticvariants associated with complex traits. Ideally one should model all covariates in unison, but most existing analysismethods for genome-wide association studies (GWAS) perform only univariate regression. Results: We extend and efficiently implement iterative hard thresholding (IHT) for multiple regression, treating all SNPssimultaneously. Our extensions accommodate generalized linear models (GLMs), prior information on genetic variants,and grouping of variants. In our simulations, IHT recovers up to 30% more true predictors than SNP-by-SNP associationtesting, and exhibits a 2 to 3 orders of magnitude decrease in false positive rates compared to lasso regression. We also testIHT on the UK Biobank hypertension phenotypes and the Northern Finland Birth Cohort of 1966 cardiovascular phenotypes.We find that IHT scales to the large datasets of contemporary human genetics and recovers the plausible genetic variantsidentified by previous studies. Conclusions: Our real data analysis and simulation studies suggest that IHT can (a) recover highly correlated predictors,(b) avoid over-fitting, (c) deliver better true positive and false positive rates than either marginal testing or lassoregression, (d) recover unbiased regression coefficients, (e) exploit prior information and group-sparsity and (f) be usedwith biobank sized data sets. Although these advances are studied for GWAS inference, our extensions are pertinent toother regression problems with large numbers of predictors.
Corresponding Author:	Kenneth Lange, Ph. D University of California Los Angeles UNITED STATES
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	University of California Los Angeles
Corresponding Author's Secondary Institution:	
First Author:	Benjamin B. Chu
First Author Secondary Information:	
Order of Authors:	Benjamin B. Chu
	Kevin L. Keys, PhD
	Christopher A. German
	Hua Zhou, PhD
	Jin J. Zhou, PhD
	Janet S. Sinsheimer, PhD
	Kenneth Lange, Ph. D
Order of Authors Secondary Information:	
Additional Information:	

Question	Response
Are you submitting this manuscript to a special series or article collection?	No
Experimental design and statistics Full details of the experimental design and statistical methods used should be given in the Methods section, as detailed in our Minimum Standards Reporting Checklist. Information essential to interpreting the data presented should be made available in the figure legends. Have you included all the information requested in your manuscript?	Yes
Resources A description of all resources used, including antibodies, cell lines, animals and software tools, with enough information to allow them to be uniquely identified, should be included in the Methods section. Authors are strongly encouraged to cite Research Resource Identifiers (RRIDs) for antibodies, model organisms and tools, where possible. Have you included the information requested as detailed in our Minimum Standards Reporting Checklist?	Yes
Availability of data and materials All datasets and code on which the conclusions of the paper rely must be either included in your submission or deposited in publicly available repositories (where available and ethically appropriate), referencing such data using a unique identifier in the references and in the "Availability of Data and Materials" section of your manuscript.	Yes

Have you have met the above requirement as detailed in our Minimum Standards Reporting Checklist?	

UNIVERSITY OF CALIFORNIA, LOS ANGELES UCLA

BERKELEY • DAVIS • IRVINE • LOS ANGELES • MERCED • RIVERSIDE • SAN DIEGO • SAN FRANCISCO SANTA BARBARA • SANTA CRUZ

DEPARTMENT OF COMPUTATIONAL MEDICINE
DAVID GEFFEN SCHOOL OF MEDICINE AT UCLA
UCLA COMPUTATIONAL MEDICINE
621 CHARLES E YOUNG DR SOUTH
LOS ANGELES, CA, 90095-1766

July 9, 2019

Dear Editor of GigaScience,

We are pleased to resubmit our research article entitled "*Iterative Hard Thresholding in GWAS: Generalized Linear Models, Prior Weights, and Double Sparsity*" for publication as a technical note in GigaScience. The title in our previous submission was "*Multivariate GWAS: Generalized Linear Models, Prior Weights, and Double Sparsity*". This paper presents unique perspectives on analyzing genome-wide association data via iterative hard thresholding.

Our original paper, submitted last July, was rejected but invited for resubmission. We have made the following improvements to our previous submission:

- Demonstration of our software on the UK biobank data.
- Modified doubly sparse group projects to handle varying group sizes.
- Updated some of our results as suggested by the reviewers. Our full response to their critique is also attached.
- Made Project.toml and Manifest.toml files available, which allows other researchers to completely reproduce the computing environment used in the manuscript. This is common practice for Julia users and has a similar effect as a Docker container.
- Extended our software to Windows users, in addition to Linux and Mac users.

Our collaborators, Dr. Hua Zhou, Dr. Jin Zhou, and Christopher German, contributed many new features and are therefore included as co-authors on this version.

Aims and scope. In this paper, we demonstrate the statistical virtues of our algorithm over its main competitors, lasso regression and marginal testing, using various simulations. We validate our results by applying our algorithms to the UK Biobank and NFBC dataset. Our results are comparable to many previous studies. These efforts culminate in a Julia package **MendelIHT.jl**. Please note that we share GigaScience's vision for reproducible research. Our package, freely available on Github[1] comes with detailed documentations[2], numerous examples, relevant source code, and .toml files that reproduces the computing environment used in our paper. Reproduction of most of our analyses requires nothing more

[1] https://github.com/OpenMendel/MendelIHT.jl
[2] https://openmendel.github.io/MendelIHT.jl/latest/

than package installation and running **Jupyter notebooks**, which automatically generate data, run the algorithms, display raw results, and compute summary statistics.

Notes on data access. All simulated data and subsequent analyses are available on our Github page. Our human genome data were generated by the Northern Finland Birth Cohort of 1966 and the UK Biobank. They are available for researchers on application.

Other confirmations. The authors declare no competing interests. All authors have approved the manuscript for submission. The content of this paper has not been published or submitted for publication elsewhere.

Sincerely,

Kenneth Lange

Janet Sinsheimer PhD
Professor of Human Genetics, Biomathematics
David Geffen School of Medicine at UCLA
Professor of Biostatistics
UCLA Fielding School of Public Health
Email: jsinshei@ucla.edu

Kenneth Lange PhD
Rosenfeld Professor of Computational Genetics
Professor of Computational Medicine, Human Genetics, and Statistics
David Geffen School of Medicine
Phone: 310-206-8076
Email: klange@ucla.edu

GigaScience, 2017, 1–14

doi: xx.xxxx/xxxx
Manuscript in Preparation
Technical Note

TECHNICAL NOTE

Iterative Hard Thresholding in GWAS: Generalized Linear Models, Prior Weights, and Double Sparsity

Benjamin B. Chu[1], Kevin L. Keys[2], Christopher A. German[3], Hua Zhou[3], Jin J. Zhou[4], Janet S. Sinsheimer[1,5,*] and Kenneth Lange[1,5,†]

[1]Department of Computational Medicine, David Geffen School of Medicine at UCLA, Los Angeles, USA and [2]Department of Medicine, University of California, San Francisco, USA and [3]Department of Biostatistics, Fielding School of Public Health at UCLA, USA and [4]Division of Epidemiology and Biostatistics, University of Arizona, Tucson, AZ 85724, USA and [5]Department of Human Genetics, David Geffen School of Medicine at UCLA, Los Angeles, USA

[*]Corresponding author. Email: jsinshei@ucla.edu
[†]Corresponding author. Email: klange@ucla.edu

Abstract

Background: Consecutive testing of single nucleotide polymorphisms (SNPs) is usually employed to identify genetic variants associated with complex traits. Ideally one should model all covariates in unison, but most existing analysis methods for genome-wide association studies (GWAS) perform only univariate regression.
Results: We extend and efficiently implement iterative hard thresholding (IHT) for multiple regression, treating all SNPs simultaneously. Our extensions accommodate generalized linear models (GLMs), prior information on genetic variants, and grouping of variants. In our simulations, IHT recovers up to 30% more true predictors than SNP-by-SNP association testing, and exhibits a 2 to 3 orders of magnitude decrease in false positive rates compared to lasso regression. We also test IHT on the UK Biobank hypertension phenotypes and the Northern Finland Birth Cohort of 1966 cardiovascular phenotypes. We find that IHT scales to the large datasets of contemporary human genetics and recovers the plausible genetic variants identified by previous studies.
Conclusions: Our real data analysis and simulation studies suggest that IHT can (a) recover highly correlated predictors, (b) avoid over-fitting, (c) deliver better true positive and false positive rates than either marginal testing or lasso regression, (d) recover unbiased regression coefficients, (e) exploit prior information and group-sparsity and (f) be used with biobank sized data sets. Although these advances are studied for GWAS inference, our extensions are pertinent to other regression problems with large numbers of predictors.

Key words: GWAS; multiple regression; high dimensional inference; iterative hard thresholding; biobank

Introduction

In genome-wide association studies (GWAS), modern genotyping technology coupled with imputation algorithms can produce an $n \times p$ genotype matrix $\mathbf{X}$ with $n \approx 10^6$ subjects and $p \approx 10^7$ genetic predictors [1, 2]. Data sets of this size require hundreds of gigabytes of disk space to store in compressed form. Decompressing data to floating point numbers for statistical analyses leads to matrices too large to fit into standard computer memory. The computational burden of dealing with massive GWAS datasets limits statistical analysis and interpretation. This paper discusses and extends a class of algorithms capable of meeting the challenge of multiple regression models with modern GWAS data scales.

Traditionally, GWAS analysis has focused on SNP-by-SNP (single nucleotide polymorphism) association testing [1, 3],

Compiled on: November 19, 2019.
Draft manuscript prepared by the author.

Key Points

- Single-SNP association testing assumes all SNPs have independent effects. IHT and lasso regression capture the effect of each SNP adjusted for all other SNPs.
- Shrinkage caused by lasso regression leaves a lot of trait variance unexplained. The variance gap is filled by false positives.
- IHT achieves more precise parameter estimates and better model selection than lasso regression.
- We extend IHT from ordinary linear regression to generalized linear regression.
- We also show how to include weights in IHT and perform doubly sparse regression (a limited number of SNP groups and a limited number of SNPs per group).

with a p-value computed for each SNP via linear regression. This approach enjoys the advantages of simplicity, interpretability, and a low computational complexity of $\mathcal{O}(np)$. Furthermore, marginal linear regressions make efficient use of computer memory, since computations are carried out on genotype *vectors* one at a time, as opposed to running on the full genotype *matrix* in multiple regression. Some authors further increase association power by reframing GWAS as a linear mixed model problem and proceeding with variance component selection [4, 5]. These advances remain within the scope of marginal analysis.

Despite their numerous successes [2], marginal regression is less than ideal for GWAS. It implicitly assumes that all SNPs have independent effects. In contrast, multiple regression can in principle model the effect of all SNPs simultaneously. This approach captures the biology behind GWAS more realistically because traits are usually determined by multiple SNPs acting in unison. Marginal regression selects associated SNPs one by one based on a pre-set threshold. Given the stringency of the p-value threshold, marginal regression can miss many causal SNPs with low effect sizes. As a result, heritability is underestimated. When $p \gg n$, one usually assumes that the number of variants k associated with a complex trait is much less than n. If this is true, we can expect multiple regression models to perform better because it a) offers better outlier detection [6] and better prediction, b) accounts for the correlations among SNPs, and c) allows investigators to model interactions. Of course, these advantages are predicated on finding the truly associated SNPs.

Adding penalties to the loss function is one way of achieving parsimony in multiple regression. The lasso [7, 8] is the most popular model selection device in current use. The lasso model selects non-zero parameters by minimizing the criterion

$$f(\beta) \quad = \quad \ell(\beta) + \lambda\|\beta\|_1,$$

where $\ell(\beta)$ is a convex loss, λ is a sparsity tuning constant, and $\|\beta\|_1 = \sum_j |\beta_j|$ is the ℓ_1 norm of the parameters. The lasso has the virtues of preserving convexity and driving most parameter estimates to 0. Minimization can be conducted efficiently via cyclic coordinate descent [9, 10]. The magnitude of the nonzero tuning constant λ determines the number of predictors selected.

Despite its widespread use, the lasso penalty has some drawbacks. First, the ℓ_1 penalty tends to shrink parameters toward 0, sometimes severely so. Second, λ must be tuned to achieve a given model size. Third, λ is chosen by cross-validation, a costly procedure. Fourth and most importantly, the shrinkage caused by the penalty leaves a lot of unexplained trait variance, which tends to encourage too many false positives to enter the model ultimately identified by cross-validation.

Inflated false positive rates can be mitigated by substituting nonconvex penalties for the ℓ_1 penalty. For example, the minimax concave penalty (MCP) [11]

$$\lambda p(\beta_j) \quad = \quad \lambda \int_0^{|\beta_j|} \left(1 - \frac{s}{\lambda\gamma}\right)_+ ds$$

starts out at $\beta_j = 0$ with slope λ and gradually transitions to a slope of 0 at $\beta_j = \lambda\gamma$. With minor adjustments, the coordinate descent algorithm for the lasso carries over to MCP penalized regression [12, 13]. Model selection is achieved without severe shrinkage, and inference in GWAS improves [14]. However, in our experience its false negative rate is considerably higher than IHT's rate [15]. A second remedy for the lasso, stability selection, weeds out false positives by looking for consistent predictor selection across random halves of the data [16]. However, it is known to be under-powered for GWAS compared to standard univariate selection [17].

In contrast, iterative hard thresholding (IHT) minimizes a loss $\ell(\beta)$ subject to the nonconvex sparsity constraint $\|\beta\|_0 \leq k$, where $\|\beta\|_0$ counts the number of non-zero components of β [18, 19, 20]. Figure 1 explains graphically how the ℓ_0 penalty reduces the bias of the selected parameters. In the figure λ, γ, and k are chosen so that the same range of β values are sent to zero. To its detriment, the lasso penalty shrinks all β's, no matter how large their absolute values. The nonconvex MCP penalty avoids shrinkage for large β's but exerts shrinkage for intermediate β's. IHT, which is both nonconvex and discontinuous, avoids shrinkage altogether. For GWAS, the sparsity model-size constant k also has a simpler and more intuitive interpretation than the lasso tuning constant λ. Finally, both false positive and false negative rates are well controlled. Balanced against these advantages is the loss of convexity in optimization and concomitant loss of computational efficiency. In practice, the computational barriers are surmountable and are compensated by the excellent results delivered by IHT in high-dimensional regression problems such as multiple GWAS regression.

[INSERT FIGURE 1 HERE]

This article has four interrelated goals. First, we extend IHT to generalized linear models. These models encompass most of applied statistics. Previous IHT algorithms focused on normal or logistic sparse regression scenarios. Our software can also perform sparse regression under Poisson and negative binomial response distributions and can be easily extended to other GLM distributions as needed. The key to our extension is the derivation of a nearly optimal step size s for improving the loglikelihood at each iteration. Second, we introduce doubly-sparse regression to IHT. Previous authors have considered group sparsity [21]. The latter tactic limits the number of groups selected. It is also useful to limit the number of predictors selected per group. Double sparsity strikes a compromise that encourages selection of correlated causative variants in linkage disequilibrium (LD). Notably, this technique generalizes group-IHT. Third, we demonstrate how to incorporate predetermined SNP weights in IHT. Our simple and interpretable weighting option allows users to introduce prior knowledge into sparse projection. Thus, one can favor predictors whose association to the response is supported by external evidence. Fourth, we present `MendelIHT.jl`: a scalable, open source, and user friendly software for IHT in the high performance programming language Julia [22].

Model Development

This section sketches our extensions of iterative hard thresholding (IHT).

IHT Background

IHT was originally formulated for sparse signal reconstruction, which is framed as sparse linear least squares regression. In classical linear regression, we are given an $n \times p$ design matrix $\mathbf{X}$ and a corresponding n-component response vector $\mathbf{y}$. We then postulate that $\mathbf{y}$ has mean $\mathrm{E}(\mathbf{y}) = \mathbf{X}\beta$ and that the residual vector $\mathbf{y} - \mathbf{X}\beta$ has independent Gaussian components with a common variance. The parameter (regression coefficient) vector β is estimated by minimizing the sum of squares $f(\beta) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|_2^2$. The solution to this problem is known as the ordinary least squares estimator and can be written explicitly as $\hat{\beta} = (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\mathbf{y}$, provided the problem is overdetermined (n > p). This paradigm breaks down in the high-dimensional regime $n \ll p$, where the parameter vector β is underdetermined. In the spirit of parsimony, IHT seeks a sparse version of β that gives a good fit to the data. This is accomplished by minimizing $f(\beta)$ subject to $\|\beta\|_0 \leq k$ for a small value of k, where $\|\cdot\|_0$ counts the number of nonzero entries of a vector. The optimization problem is formally:

$$\min \frac{1}{2}||\mathbf{y} - \mathbf{X}\beta||_2^2 \quad \text{subject to } ||\beta||_0 \leq k. \tag{1}$$

IHT abandons the explicit formula for $\hat{\beta}$ because it fails to respect sparsity and involves the numerically intractable matrix inverse $(\mathbf{X}^t\mathbf{X})^{-1}$.

IHT combines three core ideas. The first is steepest descent. Elementary calculus tells us that the negative gradient $-\nabla f(\mathbf{x})$ is the direction of steepest descent of $f(\beta)$ at $\mathbf{x}$. First-order optimization methods like IHT define the next iterate in minimization by the formula $\beta_{n+1} = \beta_n + s_n\mathbf{v}_n$, where $\mathbf{v}_n = -\nabla f(\beta_n)$ and $s_n > 0$ is some optimally chosen step size. In the case of linear regression $-\nabla f(\beta) = \mathbf{X}^t(\mathbf{y} - \mathbf{X}\beta)$. To reduce the error at each iteration, the optimal step size s_n can be selected by minimizing the second-order Taylor expansion

$$\begin{aligned} & f(\beta_n + s_n\mathbf{v}_n) \\ = \; & f(\beta_n) + s_n\nabla f(\beta_n)^t\mathbf{v}_n + \frac{s_n^2}{2}\mathbf{v}_n^t d^2 f(\beta_n)\mathbf{v}_n \\ = \; & f(\beta_n) - s_n\|\nabla f(\beta_n)\|_2^2 + \frac{s_n^2}{2}\nabla f(\beta_n)^t d^2 f(\beta_n)\nabla f(\beta_n) \end{aligned}$$

with respect to s_n. Here $d^2 f(\beta) = \mathbf{X}^t\mathbf{X}$ is the Hessian matrix of second partial derivatives. Because $f(\beta)$ is quadratic, the expansion is exact. Its minimum occurs at the step size

$$s_n = \frac{\|\nabla f(\beta_n)\|_2^2}{\nabla f(\beta_n)^t d^2 f(\beta_n)\nabla f(\beta_n)}. \tag{2}$$

This formula summarizes the second core idea.

The third component of IHT involves projecting the steepest descent update $\beta_n + s_n\mathbf{v}_n$ onto the sparsity set $S_k = \{\beta : \|\beta\|_0 \leq k\}$. The relevant projection operator $P_{S_k}(\beta)$ sets all but the k largest entries of β in magnitude to 0. In summary, IHT solves problem (1) by updating the parameter vector β according to the recipe:

$$\beta_{n+1} = P_{S_k}(\beta_n - s_n\nabla f(\beta_n))$$

with the step size given by formula (2).

Family	Mean Domain	Var(y)	$g(s)$
Normal	$\mathbb{R}$	ϕ^2	1
Poisson	$[0, \infty)$	μ	e^s
Bernoulli	$[0, 1]$	$\mu(1 - \mu)$	$\frac{e^s}{1+e^s}$
Gamma	$[0, \infty)$	$\mu^2\phi$	s^{-1}
Inverse Gaussian	$[0, \infty)$	$\mu^3\phi$	$s^{-1/2}$
Negative Binomial	$[0, \infty)$	$\mu(\mu\phi + 1)$	e^s

Table 1. Summary of mean domains and variances for common exponential distributions. In GLM, $\mu = g(\mathbf{x}^t\beta)$ denotes the mean, $s = \mathbf{x}^t\beta$ the linear responses, g is the inverse link function, and ϕ the dispersion. Except for the negative binomial, all inverse links are canonical.

An optional debiasing step can be added to improve parameter estimates. This involves replacing β_{n+1} by the exact minimum point of $f(\beta)$ in the subspace defined by the support $\{j : \beta_{n+1,j} \neq 0\}$ of β_{n+1}. Debiasing is efficient because it solves a low-dimensional problem. Several versions of hard-thresholding algorithms have been proposed in the signal processing literature. The first of these, NIHT [20], omits debaising. The rest, HTP[23], GraHTP [24], and CoSaMp [25] offer debiasing.

IHT for Generalized Linear Models

A generalized linear model (GLM) involves responses y following a natural exponential distribution with density in the canonical form

$$f(y \mid \theta, \phi) = \exp\left[\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)\right],$$

where y is the data, θ is the natural parameter, $\phi > 0$ is the scale (dispersion), and $a(\phi)$, $b(\theta)$, and $c(y, \phi)$ are known functions which vary depending on the distribution [26, 27]. Simple calculations show that y has mean $\mu = b'(\theta)$ and variance $\sigma^2 = b''(\theta)a(\phi)$; accordingly, σ^2 is a function of μ. Table 1 summarizes the mean domains and variances of a few common exponential families. Covariates enter GLM modeling through an inverse link representation $\mu = g(\mathbf{x}^t\beta)$, where $\mathbf{x}$ is a vector of covariates (predictors) and β is vector of regression coefficients (parameters). In statistical practice, data arrive as a sample of independent responses $y_1, \ldots, y_m$ with different covariate vectors $\mathbf{x}_1, \ldots, \mathbf{x}_m$. To put each predictor on an equal footing, each should be standardized to have mean 0 and variance 1. Including an additional intercept term is standard practice.

If we assemble a design matrix $\mathbf{X}$ by stacking the row vectors $\mathbf{x}_i^t$, then we can calculate the loglikelihood, score, and expected information [26, 28, 27, 29]

$$\begin{aligned} L(\beta) &= \sum_{i=1}^{n} \left[\frac{y_i\theta_i - b_i(\theta_i)}{a_i(\phi_i)} + c(y_i, \phi_i)\right] \\ \nabla L(\beta) &= \sum_{i=1}^{n} (y_i - \mu_i)\frac{g'(\mathbf{x}_i^t\beta)}{\sigma_i^2}\mathbf{x}_i \quad = \quad \mathbf{X}^t\mathbf{W}_1(\mathbf{y} - \mu) \qquad (3) \\ J(\beta) &= \sum_{i=1}^{n} \frac{1}{\sigma_i^2} g'(\mathbf{x}_i^t\beta)^2\mathbf{x}_i\mathbf{x}_i^t \quad = \quad \mathbf{X}^t\mathbf{W}_2\mathbf{X}, \end{aligned}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are two diagonal matrices. The second has positive diagonal entries; they coincide under the identity inverse link $g(s) = s$.

In the generalized linear model version of IHT, we maximize $L(\beta)$ (equivalent to minimizing $f(\beta) = -L(\beta)$) and substitute the expected information $J(\beta_n) = \mathrm{E}[-d^2L(\beta_n)]$ for $d^2f(\beta_n)$ in formula (2). This translates into the following step size in GLM estimation:

$$s_n \quad = \quad \frac{\|\nabla L(\beta_n)\|_2^2}{\nabla L(\beta_n)^t J(\beta_n)\nabla L(\beta_n)}. \qquad (4)$$

This substitution is a key ingredient of our extended IHT. It simplifies computations and guarantees that the step size is nonnegative.

Doubly Sparse Projections

The effectiveness of group sparsity in penalized regression has been demonstrated in general [30, 31] and for GWAS [32] in particular. Group IHT [21] enforces group sparsity but does not enforce within-group sparsity. In GWAS, model selection is desired within groups as well to pinpoint causal SNPs. Furthermore, one concern in GWAS is that two causative SNPs can be highly correlated with each other due to linkage disequilibrium (LD). When sensible group information is available, doubly sparse IHT encourages the detection of causative yet correlated SNPs while enforcing sparsity within groups. Here we discuss how to carry out a doubly-sparse projection that enforces both within- and between-group sparsity.

Suppose we divide the SNPs of a study into a collection G of nonoverlapping groups. Given a parameter vector β and a group $g \in G$, let β_g denote the components of β corresponding to the SNPs in g. Now suppose we want to select at most j groups and at most $\lambda_g \in \mathbb{Z}^+$ SNPs for each group g. In projecting β, the component β_i is untouched for a selected SNP i. For an unselected SNP, β_i is reset to 0. By analogy with our earlier discussion, we can define a sparsity projection operator $P_g(\beta_g)$ for each group g; $P_g(\beta_g)$ selects the λ_g most prominent SNPs in group g. The potential reduction in the squared distance offered by group g is $r_g = \|\beta_g\|_2^2 - \|P_g(\beta_g)\|_2^2$. The j selected groups are determined by selecting the j largest values of r_g. In practice we can set the sparsity level λ_g for each group high enough so that all relevant SNPs come into play. Thus, doubly-sparse IHT generalizes group-IHT. In Algorithm 1, we write $P(\beta)$ for the overall projection with the component projections $P_g(\beta_g)$ on the j selected groups and projection to zero on the remaining groups.

Prior weights in IHT

Zhou et al. [32] treat prior weights in penalized GWAS. Before calculating the lasso penalty, they multiply each component of the parameter vector β by a positive weight w_i. We can do the same in IHT before projection. Thus, instead of projecting the steepest descent step $\beta = \beta_n + s_n\mathbf{v}_n$, we project the Hadamard (pointwise) product $\mathbf{w} \circ \beta$ of β with a weight vector $\mathbf{w}$. This produces a vector with a sensible support S. The next iterate β_{n+1} is defined to have support S and to be equal to $\beta_n + s_n\mathbf{v}_n$ on S.

In GWAS, weights can and should be informed by prior biological knowledge. A simple scheme for choosing nonconstant weights relies on minor allele frequencies. For instance, Zhou et al. [33] assign SNP i with minor allele frequency p_i the weight $w_i = 1/\sqrt{2p_i(1-p_i)}$. Giving rare SNPs greater weight in this fashion is most appropriate for traits under strong negative selection [34, 35]. Alternatively, our software permits users to assign weights geared to specific pathway and gene information.

de Lamare et al. [36] incorporate prior weights into IHT by adding an element-wise logarithm of a weight vector $\mathbf{q}$ before projection. The weight vector $\mathbf{q}$ is updated iteratively and requires two additional tuning constants that in practice are only obtained through cross validation. Our weighting scheme is simpler, more computationally efficient, and more interpretable.

Algorithm Summary

The final algorithm combining doubly sparse projections, prior weight scaling, and debiasing is summarized in Algorithm 1.

Algorithm 1: Iterative hard-thresholding

```
   Input  : Design matrix $\mathbf{X}$, response vector $\mathbf{y}$, membership
            vector $\mathbf{g}$, weight vector $\mathbf{w}$, max number of
            groups $j$, and overall sparsity projection $P(\beta)$.
1  Initialize: $\beta \equiv \mathbf{0}$.
2  while not converged do
3      Calculate: score = $\mathbf{v}$, Fisher information matrix = $\mathbf{J}$,
        and step size = $s = \frac{\mathbf{v}^t\mathbf{v}}{\mathbf{v}^t\mathbf{J}\mathbf{v}}$
4      Ascent direction with scaling: $\tilde{\beta} = \mathbf{w} \circ (\beta_n + s\mathbf{v})$
5      Project to sparsity: $\tilde{\beta} = P\left(\tilde{\beta}\right)$ ./w (where ./ is
        elementwise division)
6      while $L(\tilde{\beta}) \leq L(\beta_n)$, backtrack $\leq 5$ do
7          $s = s/2$
8          Redo lines 4 to 5
9      end
10     (Optional) Debias: Let $F = \mathrm{supp}(\tilde{\beta})$, compute
        $\hat{\beta} = \mathrm{argmax}_{\{\beta:\beta \text{ restricted to } F\}} L(\beta)$
11     Accept proposal: $\beta_{n+1} = \hat{\beta}$
12 end
   Output : $\beta$ with $j$ active groups and $\lambda_g$ active predictors
            for group $g$
```

Results

Readers can reproduce our results by accessing the software, documentation, and Jupyter notebooks at:

https://github.com/OpenMendel/MendelIHT.jl

Scalability of IHT

To test the scalability of our implementation, we ran IHT on $p = 10^6$ SNPs for sample sizes $n = 10{,}000, 20{,}000, \ldots, 120{,}000$ with five independent replicates per n. All simulations rely on a true sparsity level of $k = 10$. Based on an Intel-E5-2670 machine with 63GB of RAM and a single 3.3GHz processor, Figure 2 plots the IHT median CPU time per iteration, median iterations to convergence, and median memory usage under Gaussian, logistic, Poisson, and negative binomial models. The largest matrix simulated here is 30GB in size and can still fit into our personal computer's memory. Of course, it is possible to test even larger sample sizes using cloud or cluster resources, which are often needed in practice.

The formation of the vector μ of predicted values requires only a limited number of nonzero regression coefficients. Consequently, the computational complexity of this phase of IHT is relatively light. In contrast, calculation of the Fisher score (gradient) and information (expected negative Hessian) depend on the entire genotype matrix $\mathbf{X}$. Fortunately, each of the np entries of $\mathbf{X}$ can be compressed to 2 bits. Figure 2b and d show that IHT memory demands beyond storing $\mathbf{X}$ never exceeded a few gigabytes. Figure 2a and c show that IHT run time per iteration increases linearly in problem size n. Similarly, we expect increasing p will increase run time linearly, since the bottleneck of IHT is the matrix-vector multiplication step in computing the gradient, which scales as $O(np)$. Debiasing increases run time per iteration only slightly. Except for negative binomial responses, debiasing is effective in reducing the number of iterations required for convergence and hence overall run time.

[INSERT FIGURE 2 HERE]

Cross Validation in Model Selection

In actual studies, the true number of genetic predictors k_{true} is unknown. This section investigates how q-fold cross-validation can determine the best model size on simulated data. Under normal, logistic, Poisson, and negative binomial models, we considered 50 different combinations of $\mathbf{X}$, $\mathbf{y}$, and β_{true} with $k_{\text{true}} = 10$, $n = 5000$ samples, and $p = 50,000$ SNPs fixed in all replicates. Here, k_{true} is chosen so that it is closer to our NFBC and UK Biobank results. On these data sets we conducted 5-fold cross validation across 20 model sizes k ranging from 1 to 20. Figure 3 plots deviance residuals on the holdout dataset for each of the four GLM responses (mean squared error in the case of normal responses) and the best estimate $\hat{k}$ of k_{true}.

Figure 3 shows that k_{true} can be effectively recovered by cross validation. In general, prediction error starts off high where the proposed sparsity level k severely underestimates k_{true} and plateaus when k_{true} is reached (Figure 3a-d). Furthermore, the estimated sparsity $\hat{k}$ for each run is narrowly centered around $k_{\text{true}} = 10$ (Figure 3e-f). In fact, $|\hat{k} - k_{\text{true}}| \leq 4$ always holds. When $\hat{k}$ exceeds k_{true}, the estimated regression coefficients for the false predictors tend to be very small. In other words, IHT is robust to overfitting, in contrast to lasso penalized regression. We see qualitatively similar results when k_{true} is large. This proved to be the case in our previous paper [15] for Gaussian models with $k_{\text{true}} \in \{100, 200, 300\}$.

[INSERT FIGURE 3 HERE]

Comparing IHT to Lasso and Marginal Tests in Model Selection

Comparison of the true positive and false positive rates of IHT and its main competitors is revealing. For lasso regression we use the `glmnet` implementation of cyclic coordinate descent [9, 37, 10] (v2.0-16 implemented in `R 3.5.2`); for marginal testing we use the beta version of `MendelGWAS` [38]. As explained later, Poisson regression is supplemented by zero-inflated Poisson regression implemented under the `pscl` [39] (v1.5.2) package of `R`. Unfortunately, `glmnet` does not accommodate negative binomial regression. Because both `glmnet` and `pscl` operate on floating point numbers, we limit our comparisons to small problems with 1000 subjects, 10,000 SNPs, 50 replicates, and $k = 10$ causal SNPs. IHT performs model selection by 3-fold cross validation across model sizes ranging from 1 to 50. This range is generous enough to cover the models selected by lasso regression. We adjust for multiple testing in the marginal case test by applying a p-value cutoff of 5×10^{-6}.

Table 2. IHT achieves the best balance of false positives and true positives compared to lasso and marginal (single-snp) regression.

	Normal	Logistic	Poisson	Neg Bin
IHT TP	8.84	6.28	7.2	9.0
IHT FP	0.02	0.1	1.28	0.98
Lasso TP	9.52	8.16	9.28	NA
Lasso FP	31.26	45.76	102.24	NA
Marginal TP	7.18	5.76	9.04 (5.94)	5.98
Marginal FP	0.06	0.02	1527.9 (0.0)	0.0

TP = true positives, FP = false positives. There are $k = 10$ causal SNPs. Best model size for IHT and lasso were chosen by cross validation. () = zero-inflated Poisson regression.

Table 2 demonstrates that IHT achieves the best balance between maximizing true positives and minimizing false positives. IHT finds more true positives than marginal testing and almost as many as lasso regression. IHT also finds far fewer false positives than lasso regression. Poisson regression is exceptional in yielding an excessive number of false positives in marginal testing. A similar but less extreme trend is observed for lasso regression. The marginal false positive rate is reduced by switching to zero-inflated Poisson regression. This alternative model is capable of handling overdispersion due an excess of 0 values. Interestingly, IHT rescues the Poisson model by accurately capturing the simultaneous impact of multiple predictors.

Reconstruction Quality for GWAS Data

Table 3 demonstrates that IHT estimates show little bias. These trends hold with or without debiasing as described earlier. The proportion of variance explained is approximately the same in both scenarios. The displayed values are the averaged estimated β's, computed among the SNPs actually found. As expected, lasso estimates show severe shrinkage compared to IHT. However, as the magnitude of β_{true} falls, IHT estimates show an upward absolute bias, consistent with the winner's curse phenomenon. When sample sizes are small, small effect sizes make most predictors imperceptible amid the random noise. The winner's curse operates in this regime and cannot be eliminated by IHT. Lasso's strong shrinkage overwhelms the bias of the winner's curse and yields estimates smaller than true values.

The results displayed in Table 3 reflect $n = 5,000$ subjects, $p = 10,000$ SNPs, 100 replicates, and a sparsity level k fixed at its true value $k_{\text{true}} = 10$. The λ value for lasso is chosen by cross validation. To avoid data sets with monomorphic SNPs, the minimum minor allele frequency (maf) is set at 0.05.

Correlated Covariates and Doubly Sparse Projections

Next we study how well IHT works on correlated data and whether doubly-sparse projection can enhance model selection. Table 4 shows that, in the presence of extensive LD, IHT performs reasonably well even without grouping information. When grouping information is available, group IHT enhances model selection. The results displayed in Table 4 reflect

Table 3. IHT coefficient estimates compared to lasso regression coefficient estimates.

β_{true}	$\beta^{\text{Normal}}_{\text{IHT}}$	$\beta^{\text{Normal}}_{\text{lasso}}$	$\beta^{\text{Logistic}}_{\text{IHT}}$	$\beta^{\text{Logistic}}_{\text{lasso}}$	$\beta^{\text{Poisson}}_{\text{IHT}}$	$\beta^{\text{Poisson}}_{\text{lasso}}$	$\beta^{\text{Neg Bin}}_{\text{IHT}}$	$\beta^{\text{Neg Bin}}_{\text{lasso}}$
0.50	0.499 ± 0.015	0.448 ± 0.016	0.504 ± 0.029	0.379 ± 0.030	0.498 ± 0.011	0.463 ± 0.016	0.502 ± 0.013	NA
0.25	0.250 ± 0.012	0.199 ± 0.012	0.256 ± 0.029	0.142 ± 0.028	0.249 ± 0.011	0.211 ± 0.013	0.248 ± 0.014	NA
0.10	0.096 ± 0.014	0.045 ± 0.014	0.128 ± 0.016	0.020 ± 0.015	0.099 ± 0.013	0.055 ± 0.014	0.101 ± 0.014	NA
0.05	0.062 ± 0.007	0.010 ± 0.008	0.114 ± 0.011	0.010 ± 0.011	0.057 ± 0.008	0.013 ± 0.009	0.063 ± 0.008	NA
0.03	0.057 ± 0.005	0.008 ± 0.005	NaN	NaN	0.050 ± 0.005	0.004 ± 0.004	0.060 ± 0.006	NA

Displayed coefficients are average fitted valued ± one standard error for the discovered predictors. NaN = insufficient sample size for detection. NA = `glmnet` does not support negative binomial lasso regression.

Table 4. Doubly-sparse IHT enhances model selection on simulated data.

	Ungrouped-IHT		Grouped-IHT	
	TP	FP	TP	FP
Normal	11.1 ± 1.9	3.9 ± 1.9	12.2 ± 2.0	2.8 ± 2.0
Logistic	3.8 ± 1.6	11.2 ± 1.6	7.7 ± 2.2	7.3 ± 2.2
Poisson	11.5 ± 2.2	3.5 ± 2.2	12.4 ± 1.7	2.6 ± 1.7
Neg Bin	11.0 ± 2.1	4.0 ± 2.1	12.4 ± 1.6	2.6 ± 1.6

TP = true positives, FP = false positives, ± 1 standard error. There are 15 causal SNPs in 5 groups, each containing $k \in= \{1, 2, ...5\}$ SNPs.

Table 5. Doubly sparse IHT is comparable to regular IHT on NFBC dataset using arbitrary groups

	Ungrouped-IHT		Grouped-IHT	
	TP	FP	TP	FP
Normal	17.0 ± 1.2	2.0 ± 1.2	17.0 ± 1.4	2.1 ± 1.4
Logistic	15.7 ± 1.5	3.3 ± 1.5	15.8 ± 1.6	3.2 ± 1.6
Poisson	17.1 ± 1.3	1.9 ± 1.3	17.0 ± 1.4	2.0 ± 1.4
Neg Bin	17.2 ± 1.5	1.8 ± 1.5	17.0 ± 1.5	2.1 ± 1.5

TP = true positives, FP = false positives, ± 1 standard error. There are 19 causal SNPs in 18 groups of various size. Simulation was carried out on the first 30,000 SNPs of the NFBC1966 [40] dataset.

Table 6. Weighted IHT enhances model selection.

	Unweighted-IHT		Weighted-IHT	
	TP	FP	TP	FP
Normal	9.2 ± 0.4	0.8 ± 0.4	9.4 ± 0.5	0.6 ± 0.5
Logistic	7.3 ± 0.6	2.7 ± 0.6	8.0 ± 0.6	2.0 ± 0.6
Poisson	8.0 ± 0.6	2.0 ± 0.6	8.3 ± 0.6	1.7 ± 0.6
Neg Bin	9.2 ± 0.5	0.8 ± 0.5	9.4 ± 0.5	0.6 ± 0.5

TP = true positives, FP = false positives, ± 1 standard error. The true number of SNPs is $k = 10$.

$n = 1,000$ samples, $p = 10,000$ SNPs, and 100 replicates. Each SNP belongs to 1 of 500 disjoint groups containing 20 SNPs each; $j = 5$ distinct groups are each assigned $1, 2, ..., 5$ causal SNPs with effect sizes randomly chosen from $\{-0.2, 0.2\}$. In all there 15 causal SNPs. For grouped-IHT, we assume perfect group information. That is, groups containing $1 \sim 5$ causative SNPs are assigned $\lambda_g \in \{1, 2, ..., 5\}$. The remaining groups are assigned $\lambda_g = 1$. As described in the Methods Section, the simulated data show LD within each group, with the degree of LD between two SNPs decreasing as their separation increases. Although the conditions of this simulation are somewhat idealized, they mimic what might be observed if small genetic regions of whole exome data were used with IHT.

We repeated this examination of doubly sparse projection for the first $30,000$ SNPs of the NFBC1966 [40] data for all samples passing the quality control measures outlined in our Methods Section. We arbitrarily assembled 2 large groups with 2000 SNPs, 5 medium groups with 500 SNPs, and 10 small groups with 100 SNPs, representing genes of different length. The remaining SNPs are lumped into a final group representing non-coding regions. In all there are 18 groups. Since group assignments are essentially random beyond choosing neighboring SNPs, this example represents the worse case scenario of a relatively sparse marker map with undifferentiated SNP groups. We randomly selected 1 large group, 2 medium groups, and 3 small groups to contain 5, 3, and 2 causal SNPs, respectively. The non-coding region harbors 2 causal SNPs. In all there are 19 causal SNPs. Effect sizes were randomly chosen to be −0.2 or 0.2. We ran 100 independent simulation studies under this setup, where the large, medium, small, and non-coding groups are each allowed 5, 3, 2, and 2 active SNPs. The results are displayed in Table 5. We find that even in this worse case scenario where group information is completely lacking that grouped IHT does no worse than ungrouped IHT.

Introduction of Prior Weights

This section considers how scaling by prior weights helps in model selection. Table 6 compares weighted IHT reconstructions with unweighted reconstructions where all weights $w_i = 1$. The weighted version of IHT consistently finds approximately 10% more true predictors than the unweighted version. Here we simulated 50 replicates involving 1000 subjects, 10,000 uncorrelated variants, and $k = 10$ true predictors for each GLM. For the sake of simplicity, we defined a prior weight $w_i = 2$ for about one-tenth of all variants, including the 10 true predictors. For the remaining SNPs the prior weight is $w_i = 1$. These choices reflect a scenario where one tenth of all genotyped variants fall in a protein coding region, including the 10 true predictors, and where such variants are twice as likely to influence a trait as those falling in non-coding regions.

Hypertension GWAS in the UK Biobank

Now we test IHT on the second release of UK Biobank [41] data. This dataset contains $\sim 500,000$ samples and $\sim 800,000$ SNPs without imputation. Phenotypes are systolic blood pressure (SBP) and diastolic blood pressure (DBP), averaged over 4 or fewer readings. To adjust for ancestry and relatedness, we included the following nongenetic covariates: sex, hospital center, age, age^2, BMI, and the top 10 principal components computed with `FlashPCA2` [42]. After various quality control procedures as outlined in the Methods section, the final dataset used in our analysis contains $185,565$ samples and $470,228$ SNPs. For UK biobank analysis, we omitted debiasing, prior weighting, and doubly sparse projections.

Stage 2 Hypertension under a Logistic Model

Consistent with the clinical definition for stage 2 hypertension (S2 Hyp) [43], we designated patients as hypertensive if their SBP $\geq$ 140mmHG or DBP $\geq$ 90 mmHG. We ran 5-fold cross validated logistic model across model sizes $k = \{1, 2, \ldots, 50\}$. The work load was distributed to 50 computers, each with 5 CPU cores. Each computer was assigned one model size, and all completed its task within 24 hours. The model size that minimizes the deviance residuals is $\hat{k} = 39$. The selected predictors include the 33 SNPs listed in Table 7 and 6 non-genetic covariates: intercept, sex, age, age^2, BMI, and the fifth principal component.

Among the 33 recovered SNPs, 12 are known to be associated with elevated SBP/DBP as reported in the GWAS catalog [44]. The 12 known SNPs tend to have larger absolute effect sizes (avg 0.036) than the unknown SNPs (avg = 0.029). Importantly, IHT is able to recover two pairs of highly correlated SNPs: (rs1374264,rs1898841) and (rs7497304,rs2677738) with pairwise correlations of $r_{1,2} = 0.59$ and $r_{3,4} = 0.49$.

Systolic Blood Pressure (SBP) under a Normal Model

The diagnosis of hypertension typically involves both systolic and diastolic blood pressure measurements. For purposes of illustration, we fit a normal model to systolic blood pressure, ignoring diastolic blood pressure. In this case we ran 5-fold cross validation across model sizes $k = \{1, 2, \ldots, 30\}$. The work load was distributed to 30 computers, each with 5 CPU cores. Each computer was assigned 1 model size, and all completed its task within 24 hours. The estimated model size is now just $\hat{k} = 12$. The 6 selected SNPs are listed in Table 7. The selected non-genetic covariates are the same as those identified under the logistic model. By effect size, all 6 of the selected identified SNPs are in the top third of SNPs recovered from the logistic model used for stage 2 hypertension.

Cardiovascular GWAS in NFBC1966

We also tested IHT on data from the 1966 Northern Finland Birth Cohort (NFBC1966) [40]. Although this dataset is relatively modest with 5402 participants and 364,590 SNPs, it has two virtues. First, it has been analyzed multiple times [15, 40, 45], so comparison with earlier analysis is easy. Second, due to a population bottleneck [46], the participants' chromosomes exhibit more extensive linkage disequilibrium than is typically found in less isolated populations. Multiple regression methods, including the lasso, have been criticized for their inability to deal with the dependence among predictors induced by LD. Therefore this dataset provides an interesting test case.

High Density Lipoprotein (HDL) Phenotype

Using IHT we find previously associated SNPs as well as a few new potential associations. We model the HDL phenotype as normally-distributed and find a best model size $\hat{k} = 9$ based on 5-fold cross validation across model sizes $k = \{1, 2, \ldots, 20\}$. Without debiasing, the analysis was completed in 2 hours and 4 minutes with 30 CPU cores on a single machine. Table 8 displays the recovered predictors. SNP rs1800961 was replaced by rs7499892 with similar effect size if we add the debiasing step in obtaining the final model.

Importantly, IHT is able to simultaneously recover effects for SNPs (1) rs9261224, (2) rs6917603, and (3) rs6917603 with pairwise correlations of $r_{1,2} = 0.618$, $r_{1,3} = 0.984$, and $r_{2,3} = 0.62$. This result is achieved without grouping of SNPs, which can further increase association power. Compared with earlier analyses of these data, we find 3 SNPs that were not listed in our previous IHT paper [15], presumably due to slight algorithmic modifications. The authors of NFBC [40] found 5 SNPs associated with HDL under SNP-by-SNP testing. We did not find SNPs rs2167079 and rs255049. To date, rs255049 was replicated [45]. SNP rs2167079 has been reported to be associated with an unrelated phenotype [47]. If we repeat the analysis with HDL dichotomized into low and high HDL using a cutpoint of 60ml/DL, then we identify the 5 SNPs rs9261224, rs6917603, rs9261256, rs3764261, and rs9898058; all but one of these, SNP rs9898058, is also found under the continuous model. This SNP is not replicated in previous studies. As in the continuous model, rs6917603 has the largest effect of all the

Table 7. UK Biobank GWAS results generated by running IHT on Stage 2 Hypertension (S2 Hyp) under a logistic model and Systolic Blood Pressure (SBP) under a normal model.

Trait	SNP	Position	$\hat{\beta}$	Known?
S2 Hyp	rs16998073	81184341	−0.048	[44]
	rs17367504	11862778	0.046	[44]
	rs1173771	32815028	0.046	[44]
	rs3744760	43195981	−0.043	
	rs10895001	100533021	0.043	
	rs12258967	18727959	0.039	[44]
	rs11191580	104906211	0.039	[44]
	rs2392929	106414069	−0.039	[44]
	rs167479	11526765	0.036	[44]
	rs2274224	96039597	0.036	
	rs2293579	47440758	−0.035	
	rs34328549	7253184	0.035	
	rs73203495	11580334	−0.031	
	rs2681492	90013089	0.030	[44]
	rs10849937	111792427	0.030	
	rs13107325	103188709	0.030	[44]
	rs16982520	57758720	−0.030	[44]
	rs2923089	10357572	−0.029	
	rs72742749	32834974	0.029	
	rs805293	31688518	−0.029	
	rs11241955	127626884	0.028	
	rs1530440	63524591	0.028	[44]
	rs762551	75041917	−0.027	
	rs12901664	98338524	−0.027	
	rs35085068	23409909	−0.027	
	rs2072495	158296996	−0.027	
	rs292445	55897720	−0.026	
	rs4548577	46998512	0.026	
	rs757110	17418477	−0.025	
	rs1898841	165070207	0.022	
	rs7497304	91429176	−0.021	[44]
	rs2677738	91441673	0.021	
	rs1374264	164999883	0.020	
SBP	rs17367504	11862778	0.43	[44]
	rs16998073	81184341	−0.39	[44]
	rs12258967	18727959	0.36	[44]
	rs2681492	90013089	0.33	[44]
	rs3744760	43195981	−0.33	
	rs34328549	7253184	0.33	

Table 8. NFBC GWAS results generated by running IHT on high density lipoprotein (HDL) phenotype as a normal response and low density lipoprotein (LDL) as a binary response.

Trait	SNP	Position	$\hat{\beta}$	Known?
HDL	rs6917603	30125050	0.17	[44, 15]
	rs9261256	30129920	-0.07	[15]
	rs3764261	55550825	-0.05	[44, 40, 15]
	rs1532085	56470658	-0.04	[44, 40, 15]
	rs9261224	30121866	-0.03	
	rs7120118	47242866	-0.03	[44, 40, 15]
	rs3852700	65829359	-0.03	
	rs1800961	42475778	0.03	[44]
LDL	rs6917603	30125050	-0.05	[15, 44]
	rs646776	109620053	0.03	[40, 15, 44]

selected SNPs. Readers interested in the full result can visit our Github site.

Low Density Lipoprotein (LDL) as a Binary Response

Unfortunately we did not have access to any qualitative phenotypes for this cohort, so for purposes of illustration, we fit a logistic regression model to a derived dichotomous phenotype, high versus low levels of Low Density Lipoprotein (LDL). The original data are continuous, so we choose 145 mg/dL, the midpoint [43] between the borderline-high and high LDL cholesterol categories, to separate the two categories. This dichotomization resulted in 932 cases (high LDL) and 3961 controls (low LDL). Under 5-fold cross validation without debiasing across model sizes $k = \{1, 2, \ldots, 20\}$, we find $\hat{k} = 3$. Using 30 CPU cores, our algorithm finishes in 1 hours and 7 minutes.

Despite the loss of information inherent in dichotomization, our results are comparable to the prior results under a normal model for the original quantitative LDL phenotype. Our final model still recovers two SNP predictors with and without debiasing (Table 8). We miss all but one of the SNPs that the NFBC analysis found to be associated with LDL treated as a quantitative trait. Notably we again find an association with SNP rs6917603 that they did not report.

Discussion

Multiple regression methods like iterative hard thresholding provide a principled way of model fitting and variable selection. With increasing computing power and better software, multiple regression methods are likely to prevail over univariate methods. This paper introduces a scalable implementation of iterative hard thresholding for generalized linear models. Because lasso regression can handle group and prior weights, we have also extended IHT to incorporate such prior knowledge. When it is available, enhanced IHT outperforms standard IHT. Given its sharper parameter estimates and more robust model selection, IHT is clearly superior to lasso selection or marginal association testing in GWAS.

Although we focused our attention on GWAS, our IHT implementation accepts arbitrary numeric data and is suitable for a variety of applied statistics problems. Our real data analyses and simulation studies suggest that IHT can (a) recover highly correlated SNPs, (b) avoid over-fitting, (c) deliver better true positive and false positive rates than either marginal testing or lasso regression, (d) recover unbiased regression coefficients, and (e) exploit prior information and group-sparsity. Our Julia implementation of IHT can also exploit parallel computing strategies that scale to biobank-level data. In our opinion, the time is ripe for the genomics community to embrace multiple regression models as a supplement to and possibly a replacement of marginal analysis.

The potential applications of iterative hard thresholding reach far beyond gene mapping. Genetics and the broader field of bioinformatics are blessed with rich, ultra-high dimensional data. IHT is designed to solve such problems. By extending IHT to the realm of generalized linear models, it becomes possible to fit regression models with more exotic distributions than the Gaussian distributions implicit in ordinary linear regression. In our view IHT will eventually join and probably supplant lasso regression as the method of choice in GWAS and other high-dimensional regression settings.

Methods

Data Simulation

Our simulations mimic scenarios for a range of rare and common SNPs with or without LD. Unless otherwise stated, we designate 10 SNPs to be causal with effect sizes of $0.1, 0.2, \ldots, 1.0$.

To generate independent SNP genotypes, we first sample a minor allele frequency $\rho_j \sim \text{Uniform}(0, 0.5)$ for each SNP j. To construct the genotype of person i at SNP j, we then sample from a binomial distribution with success probability ρ_j and two trials. The vector of genotypes (minor allele counts) for person i form row $\mathbf{x}_i^t$ of the design matrix $\mathbf{X}$. To generate SNP genotypes with linkage disequilibrium, we divide all SNPs into blocks of length 20. Within each block, we first sample $x_1 \sim \text{Bernoulli}(0.5)$. Then we form a single haplotype block of length 20 by the following Markov chain procedure:

$$x_{i+1} = \begin{cases} x_i & \text{with probability } \; p \\ 1 - x_i & \text{with probability } \; 1 - p \end{cases}$$

with default $p = 0.75$. For each block we form a pool of 20 haplotypes using this procedure, ensuring every one of the 40 alleles (2 at each SNP) are represented at least once. For each person, the genotype vector in a block is formed by sampling 2 haplotypes with replacement from the pool and summing the number of minor alleles at each SNP.

Depending on the simulation, the number of subjects range from $1,000$ to $120,000$, and the number of independent SNPs range from $10,000$ to $1,000,000$. We simulate data under four GLM distributions: normal (Gaussian), Bernoulli, Poisson, and negative binomial. We generate component y_i of the response vector $\mathbf{y}$ by sampling from the corresponding distribution with mean $\mu_i = g(\mathbf{x}_i^t\beta)$, where g is the inverse link function. For normal models we assume unit variance, and for negative binomial models we assume 10 required failures. To avoid overflows, we clamp the mean $g(\mathbf{x}_i^t\beta)$ to stay within $[-20, 20]$. (See Ad Hoc Tactics for a detailed explanation). We apply the canonical link for each distribution, except for the negative binomial, where we apply the log link.

Real Data's Quality Control Procedures

UK Biobank. Following the UK biobank's own quality control procedures, we first filtered all samples for sex discordance and high heterozygosity/missingness. Second, we included only people of European ancestry and excluded first and second-degree relatives based on empiric kinship coefficients. Third, we also excluded people who had taken hypertension related medications at baseline. Finally, we only included people with $\geq 98\%$ genotyping success rate over all chromosomes and SNPs with $\geq 99\%$ genotyping success rate. Calculation of kinship coefficients and filtering were carried out via the OpenMendel modules [38] `MendelKinship` and `SnpArrays`. Remaining missing genotypes were imputed using modal genotypes at each SNP.

Northern Finland Birth Cohort. We imputed missing genotypes with Mendel [48]. Following [15], we excluded subjects with missing phenotypes, fasting subjects, and subjects on diabetes medication. We conducted quality control measures using the OpenMendel module `SnpArrays`[38]. Based on these measures, we excluded SNPs with minor allele frequency ≤ 0.01 and Hardy Weinberg equilibrium p-values $\leq 10^{-5}$. As for non-genetic predictors, we included sex (the `sexOCPG` factor defined in [40]) as well as the first 2 principal components of the genotype matrix computed via `PLINK 2.0 alpha` [49]. To put predictors, genetic and non-genetic, on an equal footing, we standardized all predictors to have mean zero and unit variance.

Linear Algebra with Compressed Genotype Files

The genotype count matrix stores minor allele counts. The PLINK genotype compression protocol [49] compactly stores the corresponding 0's, 1's, and 2's in 2 bits per SNP, achieving a compression ratio of 32:1 compared to storage as floating point numbers. For a sparsity level k model, we use OpenBLAS (a highly optimized linear algebra library) to compute predicted values. This requires transforming the k pertinent columns of $\mathbf{X}$ into a floating point matrix $\mathbf{X}_k$ and multiplying it times the corresponding entries β_k of β. The inverse link is then applied to $\mathbf{X}_k\beta_k$ to give the mean vector $\mu = g(\mathbf{X}_k\beta_k)$. In computing the GLM gradient (equation 3), formation of the vector $\mathbf{W}_1(\mathbf{y} - \mu)$ involves no matrix multiplications. Computation of the gradient $\mathbf{X}^t\mathbf{W}_1(\mathbf{y} - \mu)$ is more complicated because the full matrix $\mathbf{X}$ can no longer be avoided. Fortunately, the OpenMendel module `SnpArrays` can be invoked to perform compressed matrix times vector multiplication. Calculation of the steplength of IHT requires computation of the quadratic form $\nabla L(\beta_n)^t\mathbf{X}^t\mathbf{W}_2\mathbf{X}\nabla L(\beta_n)$. Given the gradient, this computation requires a single compressed matrix times vector multiplication. Finally, good statistical practice calls for standardizing covariates. To standardize the genotype counts for SNP j, we estimate its minor allele frequency p_j and then substitute the ratio $\frac{x_{ij}-2p_j}{\sqrt{2p_j(1-p_j)}}$ for the genotype count x_{ij} for person i at SNP j. This procedure is predicated on a binomial distribution for the count x_{ij}. Our previous paper [15] shows how to accommodate standardization in the matrix operations of IHT without actually forming or storing the standardized matrix.

Although multiplication via the OpenMendel module `SnpArrays` [38] is slower than OpenBLAS multiplication on small data sets, it can be as much as 10 times faster on large data sets. OpenBLAS has advantages in parallelization, but it requires floating point arrays. Once the genotype matrix $\mathbf{X}$ exceeds the memory available in RAM, expensive data swapping between RAM and hard disk memory sets in. This dramatically slows matrix multiplication. `SnpArrays` is less vulnerable to this hazard owing to compression. Once compressed data exceeds RAM, `SnpArrays` also succumbs to the swapping problem. Current laptop and desktop computers seldom have more than 32 GB of RAM, so we must resort to cluster or cloud computing when input files exceed 32 GB.

Computations Involving Non-genetic Covariates

Non-genetic covariates are stored as double or single precision floating point entries in an $n \times r$ design matrix $\mathbf{Z}$. To accommodate an intercept, the first column should be a vector of 1's. Let γ denote the r vector of regression coefficients corresponding to $\mathbf{Z}$. The full design matrix is the block matrix $(\mathbf{X}\,\mathbf{Z})$. Matrix multiplications involving $(\mathbf{X}\,\mathbf{Z})$ should be carried out via

$$(\mathbf{X}\,\mathbf{Z})\begin{pmatrix}\beta\\ \gamma\end{pmatrix} = \mathbf{X}\beta + \mathbf{Z}\gamma \quad \text{and} \quad (\mathbf{X}\,\mathbf{Z})^t\,\mathbf{v} = \begin{pmatrix}\mathbf{X}^t\mathbf{v}\\ \mathbf{Z}^t\mathbf{v}\end{pmatrix}.$$

Adherence to these rules ensures a low memory footprint. Multiplication involving $\mathbf{X}$ can be conducted as previously explained. Multiplication involving $\mathbf{Z}$ can revert to BLAS.

Parallel Computation

The OpenBLAS library accessed by Julia is inherently parallel. Beyond that we incorporate parallel processing in cross validation. Recall that in q-fold cross validation we separate subjects into q disjoint subsets. We then fit a training model using $q - 1$ of those subsets on all desired sparsity levels and record the mean-squared prediction error on the omitted subset. Each of the q subsets serve as the testing set exactly once. Testing error is averaged across the different folds for each sparsity levels k. The lowest average testing error determines the recommended sparsity.

`MendelIHT.jl` offers 2 parallelism strategies in cross validation. Either the q training sets are each loaded to q different CPUs where each compute and test differ sparsity levels sequentially, or each of the q training sets are cycled through sequentially and each sparsity parameter is fitted and tested in parallel. The former tactic requires enough disk space and RAM to store q different training data (where each typically require $(q - 1)/q$ GB of the full data), but offers immense parallel power because one can assign different computers to handle different sparsity levels. This tactic allows one to fit biobank scale data in less than a day assuming enough storage space and computers are available. The latter tactic requires cycling through the training sets sequentially. Since intermediate data can be deleted, the tactic only requires enough disk space and RAM to store 1 copy of the training set. `MendelIHT.jl` uses one of Julia's [22] standard library `Distributed.jl` to achieve the aforementioned parallel strategies.

Ad Hoc Tactics to Prevent Overflows

In Poisson and negative binomial regressions, the inverse link argument $\exp(\mathbf{x}_i^t\beta)$ experiences numerical overflows when the inner product $\mathbf{x}_i^t\beta$ is too large. In general, we avoid running Poisson regression when response means are large. In this regime a normal approximation is preferred. As a safety feature, `MendelIHT.jl` clamps values of $\mathbf{x}_i^t\beta$ to the interval $[-20, 20]$. Note that penalized regression suffers from the same overflow catastrophes.

Convergence and Backtracking

For each proposed IHT step we check whether the objective $L(\beta)$ increases. When it does not, we step-halve at most 5 times to restore the ascent property. Convergence is declared when

$$\frac{||\beta_{n+1} - \beta_n||_\infty}{||\beta_n||_\infty + 1} < \text{Tolerance},$$

with the default tolerance being 0.0001. The addition of 1 in the denominator of the convergence criterion guards against division by 0.

Availability of source code

Project name: MendelIHT
Project home page:
https://github.com/OpenMendel/MendelIHT.jl
Operating systems: Mac OS, Linux, Windows
Programming language: Julia 1.0, 1.2
License: MIT

The code to generate simulated data, as well as their subsequent analysis, are available in our github repository under *figures* folder. Project.toml and Manifest.toml files can be used together to instantiate the same computing environment in our paper. Notably, `MendelIHT.jl` interfaces with the OpenMendel package `SnpArrays.jl` [38] and JuliaStats's packages `Distribution.jl` [50] and `GLM.jl` [51].

Availability of supporting data and materials

The Northern Finland Birth Cohort 1966 (NFBC1966) [40] was downloaded from dbGaP under dataset accession pht002005.v1.p1. UK Biobank data are retrieved under Project ID: 48152 and 15678.

Declarations

List of abbreviations

GWAS: genome wide association studies; SNP: single nucleotide polymorphism; IHT: iterative hard threhsolding; GLM: generalized linear models; LD: linkage disequilibrium; MAF: minor allele frequency; Neg Bin: negative binomial; NFBC: northern finland birth cohort; HDL: high density lipoprotein; LDL: low density lipoprotein; SBP: systolic blood pressure; DBP: diastolic blood pressure;

Ethics, Consent for publication, competing interest

The authors declare no conflicts of interest. As described in [40], informed consent from all study subjects of NFBC1966 was obtained using protocols approved by the Ethical Committee of the Northern Ostrobothnia Hospital District.

Funding

Benjamin Chu was supported by NIH T32-HG002536 training grant and the 2018 Google Summer of Code. Kevin Keys was supported by a diversity supplement to NHLBI grant R01HL135156, the UCSF Bakar Computational Health Sciences Institute, the Gordon and Betty More Foundation grant GBMF3834, and the Alfred P. Sloan Foundation grant 2013-10-27 to UC Berkeley through the Moore-Sloan Data Sciences Environment initiative at the Berkeley Institute for Data Science (BIDS). Kenneth Lange and Hua Zhou were supported by grants from the National Human Genome Research Institute (HG006139) and the National Institute of General Medical Sciences (GM053275). Janet Sinsheimer was supported by grants from the National Institute of General Medical Sciences (GM053275), the National Human Genome Research Institute (HG009120) and the National Science Foundation (DMS-1264153). Christopher German was supported by the Burroughs Wellcome Fund Inter-school Training Program in Chronic Diseases (BWF-CHIP).

Author's Contributions

JSS, KK, KL, BC contributed to the design of the study, interpretation of results, and writing of the manuscript. BC designed and implemented the simulations and conducted the data analyses. HZ, CG, and JZ contributed to the analysis of UKBB results. KK and BC developed the software. KL and BC developed the algorithms.

Acknowledgements

We thank UCLA for providing the computing resources via the Hoffman2 cluster.

The NFBC1966 Study is conducted and supported by the National Heart, Lung, and Blood Institute (NHLBI) in collaboration with the Broad Institute, UCLA, University of Oulu, and the National Institute for Health and Welfare in Finland. This manuscript was not prepared in collaboration with investigators of the NFBC1966 Study and does not necessarily reflect the opinions or views of the NFBC1966 Study Investigators, Broad Institute, UCLA, University of Oulu, National Institute for Health and Welfare in Finland and the NHLBI.

References

1. Cantor RM, Lange K, Sinsheimer JS. Prioritizing GWAS results: a review of statistical methods and recommendations for their application. The American Journal of Human Genetics 2010;86:6–22.
2. Visscher PM, Wray NR, Zhang Q, Sklar P, McCarthy MI, Brown MA, et al. 10 years of GWAS discovery: biology, function, and translation. The American Journal of Human Genetics 2017;101:5–22.
3. Bush WS, Moore JH. Genome-wide association studies. PLoS Computational Biology 2012;8:e1002822.
4. Han B, Eskin E. Random-effects model aimed at discovering associations in meta-analysis of genome-wide association studies. The American Journal of Human Genetics 2011;88:586–598.
5. Loh PR, Tucker G, Bulik-Sullivan BK, Vilhjalmsson BJ, Finucane HK, Salem RM, et al. Efficient Bayesian mixed-model analysis increases association power in large cohorts. Nature Genetics 2015;47:284.
6. Rahman SK, Sathik MM, Kannan KS. Multiple linear regression models in outlier detection. International Journal of Research in Computer Science 2012;2(2):23.
7. Tibshirani R. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B (Methodological) 1996;p. 267–288.
8. Vattikuti S, Lee JJ, Chang CC, Hsu SD, Chow CC. Applying compressed sensing to genome-wide association studies. GigaScience 2014;3:10.
9. Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software 2010;33:1.
10. Wu TT, Lange K. Coordinate descent algorithms for lasso penalized regression. The Annals of Applied Statistics 2008;2:224–244.
11. Zhang T. Analysis of Multi-stage Convex Relaxation for Sparse Regularization. Journal of Machine Learning Research 2010;11:1081–1107.
12. Breheny P, Huang J. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. Annals of Applied Statistics 2011;5:232–253.
13. Mazumder R, Friedman JH, Hastie T. SparseNet: Coordinate Descent With Nonconvex Penalties. Journal of the American Statistical Association 2011;106:1125–1138.
14. Hoffman GE, Logsdon BA, Mezey JG. PUMA: A Unified Framework for Penalized Multiple Regression Analysis of GWAS Data. PLoS Computational Biology 2013 06;9:e1003101.
15. Keys KL, Chen GK, Lange K. Iterative hard thresholding for model selection in genome-wide association studies. Genetic Epidemiology 2017;41:756–768.
16. Meinshausen N, Bühlmann P. Stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 2010;72:417–473.
17. Alexander DH, Lange K. Stability selection for genome-wide association. Genetic Epidemiology 2011;35:722–728.
18. Beck A. Introduction to nonlinear optimization: Theory, algorithms, and applications with MATLAB, vol. 19. Siam; 2014.
19. Beck A, Teboulle M. A linearly convergent algorithm for

solving a class of nonconvex/affine feasibility problems. In: Fixed-Point Algorithms for Inverse Problems in Science and Engineering Springer; 2011.p. 33–48.

20. Blumensath T, Davies ME. Normalized iterative hard thresholding: Guaranteed stability and performance. IEEE Journal of Selected Topics in Signal Processing 2010;4:298–309.
21. Yang F, Barber RF, Jain P, Lafferty J. Selective inference for group-sparse linear models. In: Advances in Neural Information Processing Systems; 2016. p. 2469–2477.
22. Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: A fresh approach to numerical computing. SIAM Review 2017;59:65–98.
23. Foucart S. Hard thresholding pursuit: an algorithm for compressive sensing. SIAM Journal on Numerical Analysis 2011;49:2543–2563.
24. Yuan XT, Li P, Zhang T. Gradient hard thresholding pursuit. Journal of Machine Learning Research 2017;18:166–1.
25. Needell D, Tropp JA. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. Applied and Computational Harmonic Analysis 2009;26:301–321.
26. Dobson AJ, Barnett A. An introduction to generalized linear models. Chapman and Hall/CRC; 2008.
27. McCullagh P. Generalized Linear Models. Routledge; 2018.
28. Lange K. Numerical analysis for statisticians. Springer Science & Business Media; 2010.
29. Xu J, Chi E, Lange K. Generalized Linear Model Regression under Distance-to-set Penalties. In: Advances in Neural Information Processing Systems 30. Curran Associates, Inc.; 2017.p. 1385–1395.
30. Meier L, Van De Geer S, Bühlmann P. The group lasso for logistic regression. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 2008;70:53–71.
31. Friedman J, Hastie T, Tibshirani R. A note on the group lasso and a sparse group lasso. arXiv preprint arXiv:10010736 2010;.
32. Zhou H, Sehl ME, Sinsheimer JS, Lange K. Association screening of common and rare genetic variants by penalized regression. Bioinformatics 2010;26:2375.
33. Zhou H, Alexander DH, Sehl ME, Sinsheimer JS, Sobel EM, Lange K. Penalized regression for genome-wide association screening of sequence data. In: Pacific Symposium on Biocomputing World Scientific; 2011.p. 106–117.
34. Zeng J, De Vlaming R, Wu Y, Robinson MR, Lloyd-Jones LR, Yengo L, et al. Signatures of negative selection in the genetic architecture of human complex traits. Nature genetics 2018;50(5):746.
35. Schoech AP, Jordan DM, Loh PR, Gazal S, O'Connor LJ, Balick DJ, et al. Quantification of frequency-dependent genetic architectures in 25 UK Biobank traits reveals action of negative selection. Nature communications 2019;10(1):790.
36. de Lamare, Rodrigo C. Knowledge-Aided Normalized Iterative Hard Thresholding Algorithms and Applications to Sparse Reconstruction. arXiv preprint arXiv:180909281 2018;.
37. Wu TT, Chen YF, Hastie T, Sobel E, Lange K. Genome-wide association analysis by lasso penalized logistic regression. Bioinformatics 2009;25:714–721.
38. Zhou H, Sinsheimer JS, Bates DM, Chu BB, German CA, Ji SS, et al. OpenMendel: a cooperative programming project for statistical genetics. Human Genetics 2019;p. 1–11.
39. Zeileis A, Kleiber C, Jackman S. Regression Models for Count Data in R. Journal of Statistical Software 2008;27:1–25.
40. Sabatti C, Service SK, Hartikainen AL, Pouta A, Ripatti S, Brodsky J, et al. Genome-wide association analysis of metabolic traits in a birth cohort from a founder population. Nature genetics 2009;41:35.
41. Sudlow C, Gallacher J, Allen N, Beral V, Burton P, Danesh J, et al. UK BioBank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. PLoS Medicine 2015;12:e1001779.
42. Abraham G, Qiu Y, Inouye M. FlashPCA: principal component analysis of Biobank-scale genotype datasets. Bioinformatics 2017;.
43. Whelton PK, Carey RM, Aronow WS, Casey DE, Collins KJ, Himmelfarb CD, et al. 2017 ACC/AHA/AAPA/ABC/ACPM/AGS/APhA/ASH/ASPC/NMA/PCNA guideline for the prevention, detection, evaluation, and management of high blood pressure in adults: a report of the American College of Cardiology/American Heart Association Task Force on Clinical Practice Guidelines. Journal of the American College of Cardiology 2018;71(19):e127–e248.
44. MacArthur J, Bowler E, Cerezo M, Gil L, Hall P, Hastings E, et al. The new NHGRI-EBI Catalog of published genome-wide association studies (GWAS Catalog). Nucleic acids research 2016;45:D896–D901.
45. Gai L, Eskin E. Finding associated variants in genome-wide association studies on multiple traits. Bioinformatics 2018;34:i467–i474.
46. Martin AR, Karczewski KJ, Kerminen S, Kurki MI, Sarin AP, Artomov M, et al. Haplotype sharing provides insights into fine-scale population history and disease in Finland. The American Journal of Human Genetics 2018;102:760–775.
47. Melquist S, Craig DW, Huentelman MJ, Crook R, Pearson JV, Baker M, et al. Identification of a novel risk locus for progressive supranuclear palsy by a pooled genomewide scan of 500,288 single-nucleotide polymorphisms. The American Journal of Human Genetics 2007;80:769–778.
48. Lange K, Papp JC, Sinsheimer JS, Sripracha R, Zhou H, Sobel EM. Mendel: the Swiss army knife of genetic analysis programs. Bioinformatics 2013;29:1568–1570.
49. Chang CC, Chow CC, Tellier LC, Vattikuti S, Purcell SM, Lee JJ. Second-generation PLINK: rising to the challenge of larger and richer datasets. GigaScience 2015;4:7.
50. Besançon M, Anthoff D, Arslan A, Byrne S, Lin D, Papamarkou T, et al. Distributions.jl: Definition and Modeling of Probability Distributions in the JuliaStats Ecosystem. arXiv e-prints 2019 Jul;p. arXiv:1907.08611.
51. Lin D, White JM, Byrne S, Bates D, Noack A, Pearson J, et al., JuliaStats/Distributions.jl: a Julia package for probability distributions and associated functions; 2019.

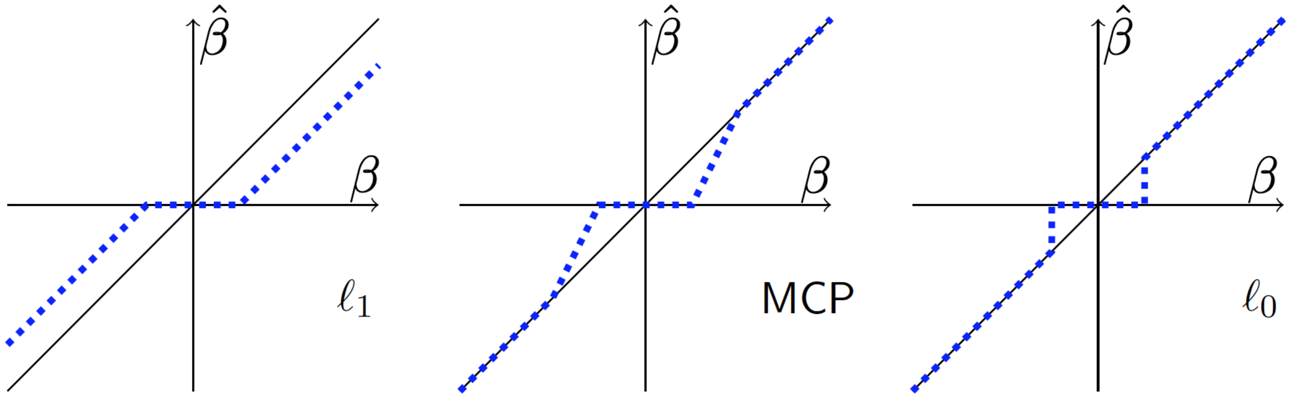
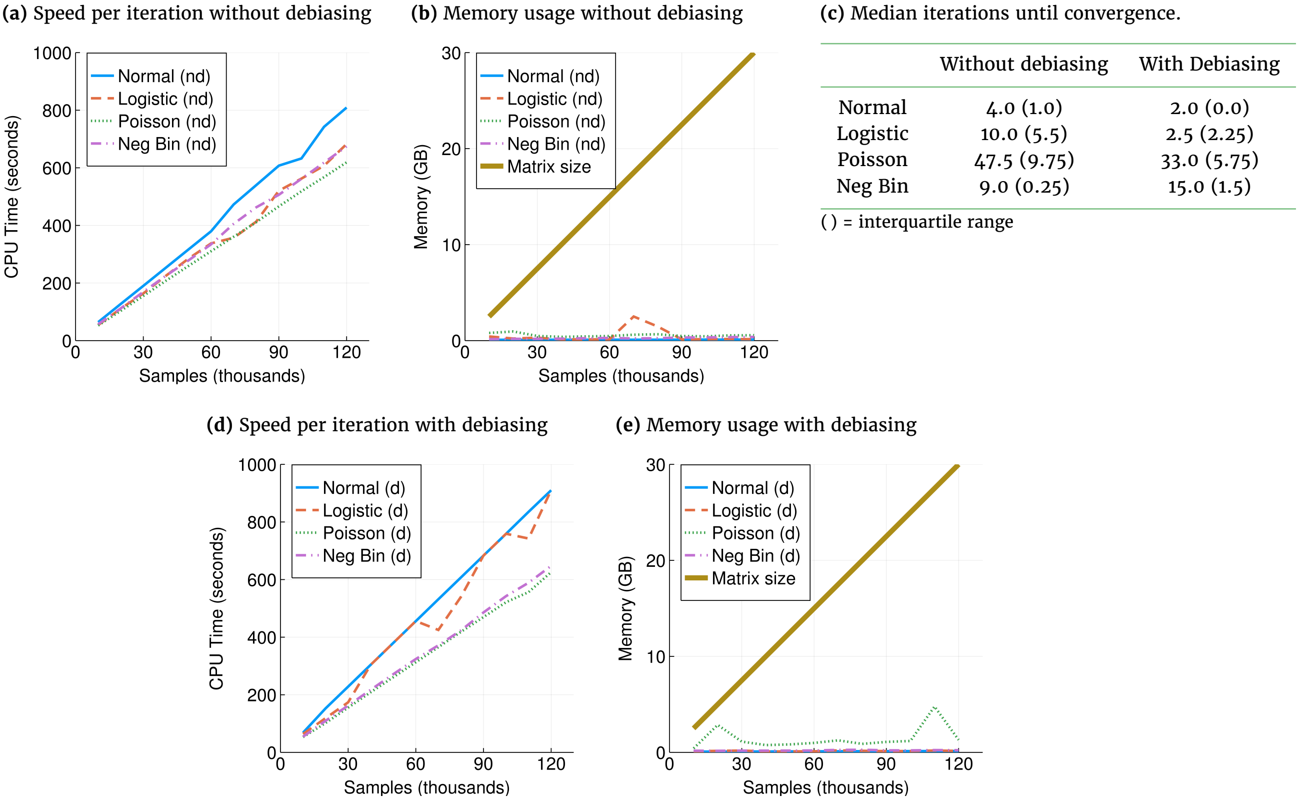


Figure 1. The ℓ_0 quasinorm of IHT enforces sparsity without shrinkage. The estimated effect size (dashed line) is plotted against its true value (diagonal line) for ℓ_1, MPC, and ℓ_0 penalties.



(a) Speed per iteration without debiasing

(b) Memory usage without debiasing

(c) Median iterations until convergence.

	Without debiasing	With Debiasing
Normal	4.0 (1.0)	2.0 (0.0)
Logistic	10.0 (5.5)	2.5 (2.25)
Poisson	47.5 (9.75)	33.0 (5.75)
Neg Bin	9.0 (0.25)	15.0 (1.5)

( ) = interquartile range

(d) Speed per iteration with debiasing

(e) Memory usage with debiasing

Figure 2. (a, d) Time per iteration scales linearly with data size. Speed is measured for compressed genotype files. On uncompressed data, all responses are roughly 10 times faster. (b, e) Memory usage scales as $\sim 2np$ bits. Note memory for each response are usages in addition to loading the genotype matrix. Uncompressed data requires 32 times more memory. (c) Debiasing reduces median iterations until convergence for all but negative binomial regression. Benchmarks were carried out on 10^6 SNPs and sample sizes ranging from 10,000 to 120,000. Hence, the largest matrix here requires 30GB and can still fit into personal computer memories.

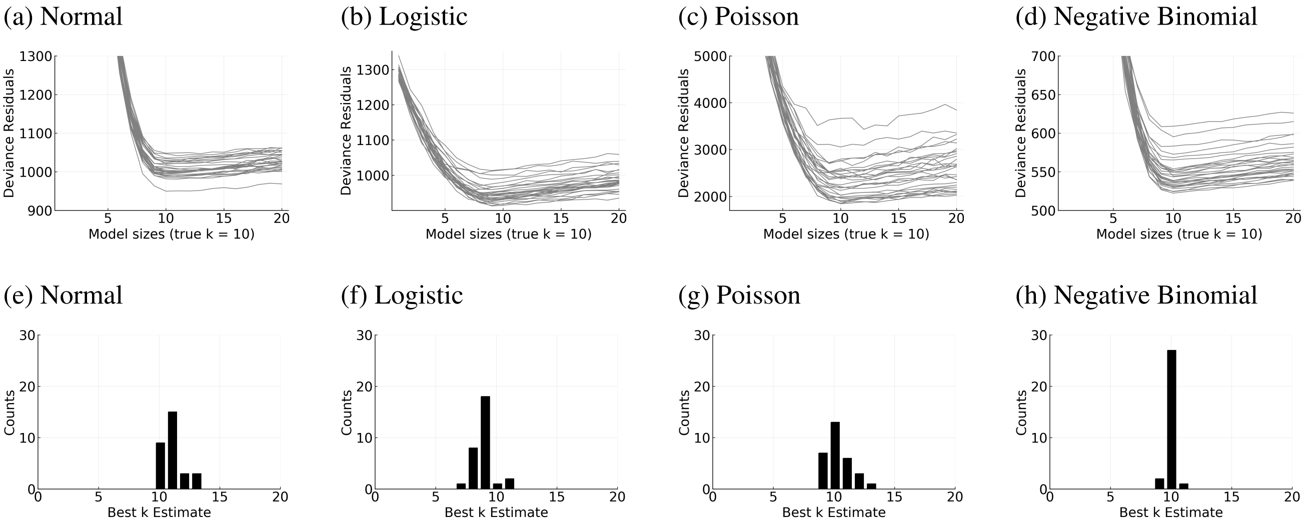


Figure 3. Five-fold cross validation results is capable of identifying the true model size k_{true}. (a–d) Deviance residuals of the testing set are minimized when the estimated model size $\hat{k} \approx k_{\text{true}}$. Each line represents 1 simulation. (e–h) $\hat{k}$ is narrowly spread around k_{true} = 10.

```
This is pdfTeX, Version 3.14159265-2.6-1.40.19 (TeX Live 2018/W32TeX)
(preloaded format=pdflatex 2018.7.12)  19 NOV 2019 13:01
entering extended mode
 restricted \write18 enabled.
 %&-line parsing enabled.
**iht_gigascience.tex
(./IHT_GigaScience.tex
LaTeX2e <2018-04-01> patch level 5

! LaTeX Error: File `oup-contemporary.cls' not found.

Type X to quit or <RETURN> to proceed,
or enter new name. (Default extension: cls)

Enter file name:
! Emergency stop.
<read *>

l.11 ^^M

*** (cannot \read from terminal in nonstop modes)


Here is how much of TeX's memory you used:
 11 strings out of 492646
 274 string characters out of 6133325
 56709 words of memory out of 5000000
 3994 multiletter control sequences out of 15000+600000
 3640 words of font info for 14 fonts, out of 8000000 for 9000
 1141 hyphenation exceptions out of 8191
 10i,0n,8p,97b,8s stack positions out of 5000i,500n,10000p,200000b,80000s
!  ==> Fatal error occurred, no output PDF file produced!
```

Click here to download Figure shrinkage2.png

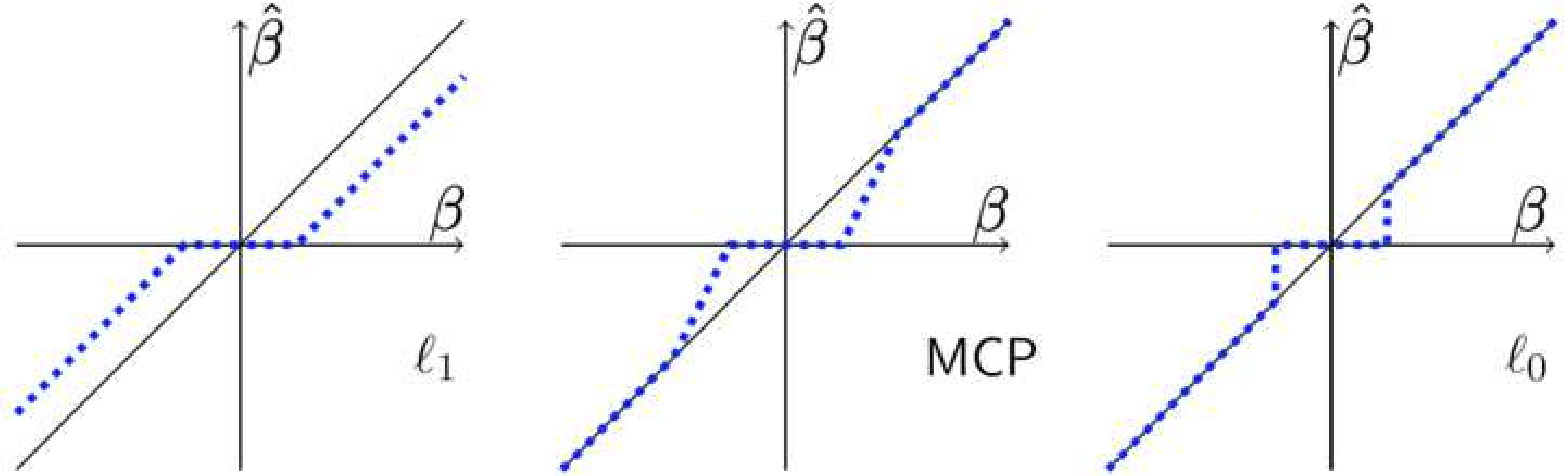
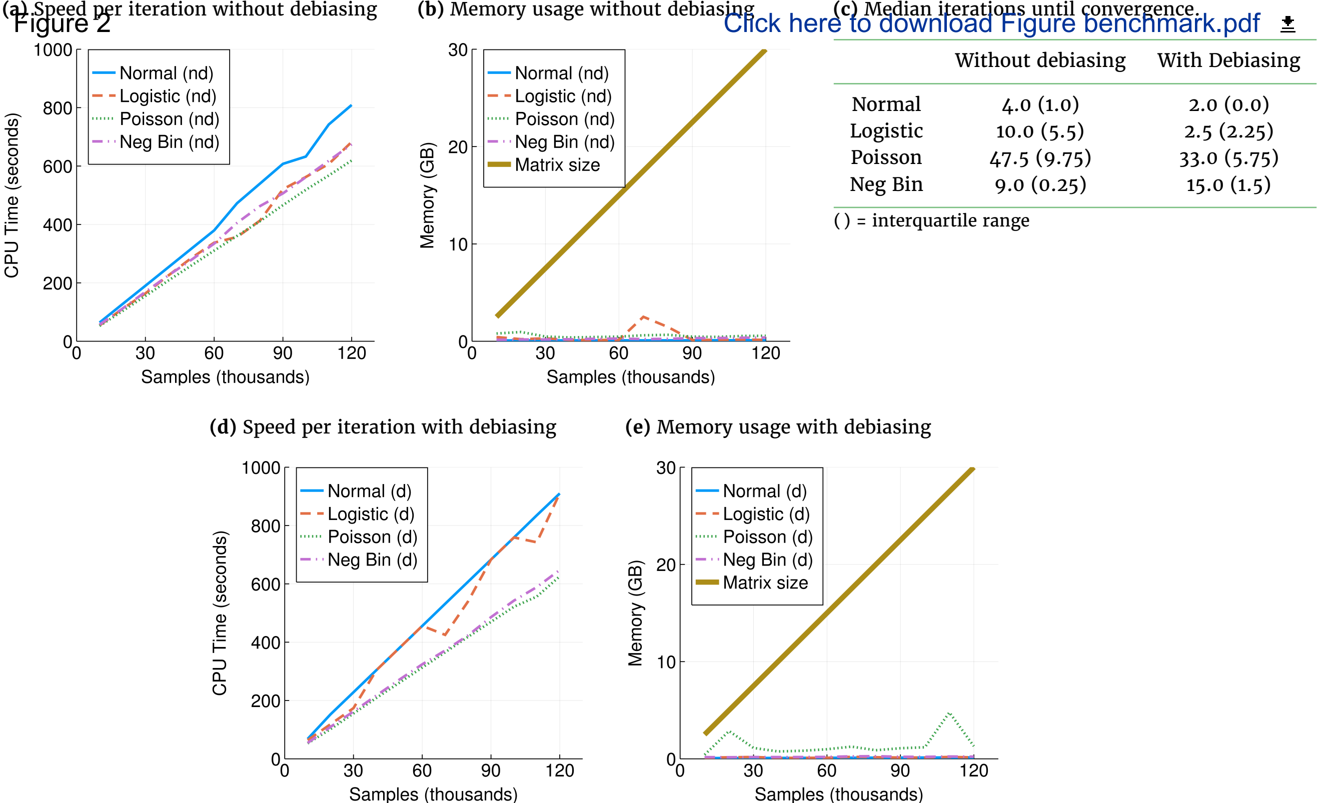


Figure 2

Click here to download Figure benchmark.pdf



(a) Speed per iteration without debiasing

(b) Memory usage without debiasing

(c) Median iterations until convergence.

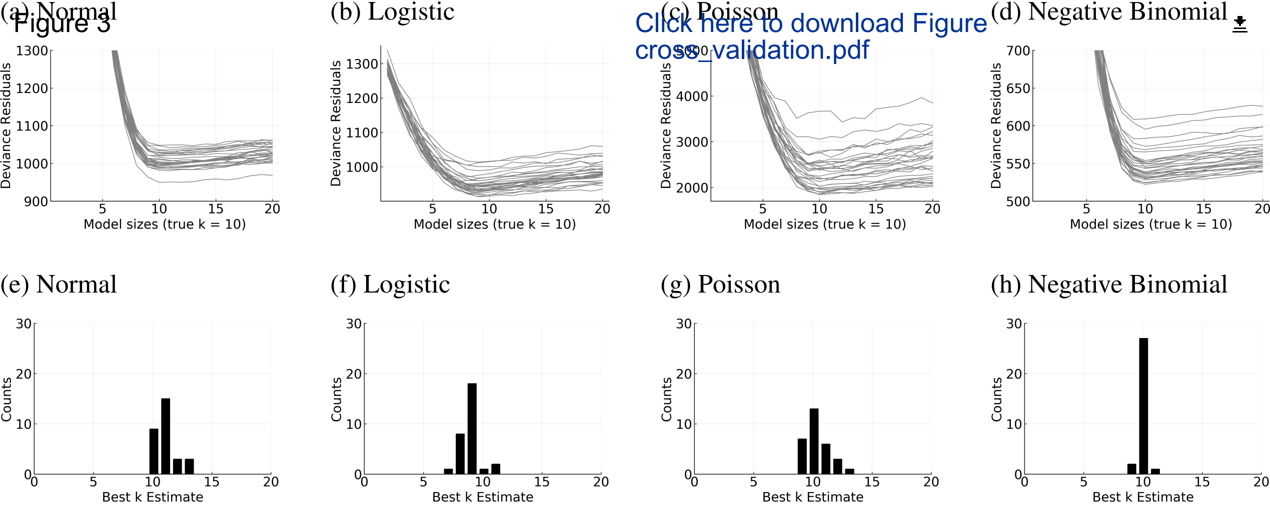
	Without debiasing	With Debiasing
Normal	4.0 (1.0)	2.0 (0.0)
Logistic	10.0 (5.5)	2.5 (2.25)
Poisson	47.5 (9.75)	33.0 (5.75)
Neg Bin	9.0 (0.25)	15.0 (1.5)

( ) = interquartile range

(d) Speed per iteration with debiasing

(e) Memory usage with debiasing

Figure 3



(a) Normal

(b) Logistic

(c) Poisson

(d) Negative Binomial

(e) Normal

(f) Logistic

(g) Poisson

(h) Negative Binomial

Responses to Reviewer Comments:

Reviewer 1

Major:

1. They start with a set of simulations with 50k unlinked markers. I am fine with this setting given the estimated number of independent segments in the human genome (Wray et al 2013 Nat Rev Genet). However, I find the number of causal variants (k=10) in their simulation is unrealistically small. Latest GWAS of large sample sizes have revealed hundreds to thousands trait-associated SNPs that are nearly independent. I suggest to add simulations with a larger k, e.g. 100, with the causal effects randomly sampled from a normal distribution and control the trait heritability, rather than giving fixed values to the causal effects.

We have two reasons for our desire to leave the number of causal variants at 10.

- We used IHT to fit 4 models on the NFBC1966 and UK Biobank datasets. In these cases, the estimated true ks by cross-validation are 39, 12, 9, and 3, which are closer to 10 than 100. Therefore, we picked k = 10 to resemble these scenarios.
- We tested cross validation for true k=100, 200, and 300 in our previous article (*Keys et al 2017 Genetic Epidemiology*). In those cases, the cross validation results were comparable to those in our current article and the size of true k does not change the results.

We have added these two explanations in the revised article.

We tried the reviewer's suggestion of simulating effect sizes randomly from a standard normal N(0, 1). Ultimately, we decided to base our simulations on fixed effect sizes as described in the article. Our reasons are:

- We want to examine IHT's performance for small and large effect sizes. It is more intuitive if we construct equidistant effect sizes as opposed to random effect sizes.
- Our phenotype data are generated from 4 distributions: normal, Bernoulli, Poisson, and negative binomial. To provide a basis for comparing their performance, it is ideal if the simulated effect sizes were drawn from the same distribution. Although using a N(0, 1) distribution would provide sufficiently large effect sizes so that model selection for logistic regression is feasible, it introduces overflow issues for Poisson and negative binomial simulation for k > 5. This problem is described in the *Methods Section*.
- If we don't insist on the same simulations for different distributions, we could use a smaller variance for Poisson and negative binomial to circumvent the overflow issues. One solution would be to simulate causal variants from N(0, 0.25). However, doing so creates a much harder regression problem because most of the causal variants would exhibit effect sizes very close to 0. This will make the proportion of successfully recovered predictors for Poisson and negative binomial IHT disproportionately low compared to normal and logistic IHT. This would be confusing and misleading for many readers.

2. They point out one major drawback of lasso is inflated false positive rates and briefly introduce two remedies for this issue, minimax concave penalty and stability selection. However, when they compare their method with lasso, it seems only the standard lasso is used. It would be interesting to compare to the lasso with these two optimisations to see if the advantage of their method still holds.

We tried stability selection in the revision process, but the result is not included in the revised article for two reasons:

- The only implementation of stability selection we found was the **stabs** package in **R** which complements with **glmnet**'s lasso implementation. Unfortunately, **stabs** consistently select either 0 variables, or the first dozen or so variables (which is not informative). This issue has been reported numerous times in the author's GitHub and we suspect it is a problem with their code. We believe it is beyond reasonable effort if we have to implement this ourselves or fix their error for them.

- It is known that stability selection is underpowered compared to marginal testing for GWAS (*Alexander et al 2010 Genetic Epidemiology*). Since we showed that IHT is superior to marginal testing, it is unlikely for stability selection to be better than IHT.

There are two reasons why we omitted the minimax concave penalty in this article.

- In our previous article (*Keys et al 2017 Genetic Epidemiology*), we already compared IHT to the minimax concave penalty for the Gaussian case. The result indicated that MCP penalty admits too many false negative in model selection. The high false negative rate will likely carry over to logistic, Poisson, or negative binomial models studied in this article.
- MCP penalty is less practical compared to IHT or the standard lasso because one has to tune 2 hyperparameters as opposed to 1. This makes MCP harder to use than either lasso or IHT.

These explanations and their related citations are included in the revised article.

3. The inflated estimate of SNP effect in Table 3 when true beta < 0.05 worries me. I thought the number in the table is the average of the beta estimate across simulation replicates regardless of the p-value of the beta. Then there shouldn't be winner's curse? In contrast, I thought the small effect is supposed to be shrunk more heavily toward zero relative to the large effect according to Figure 1. So, I don't understand where the inflation comes from. In addition, it will be helpful to assess the importance of the bias if they quantify the genetic variance explained by the SNP with such a true beta value.

We thank the reviewer for reminding us that the estimates of the selected betas are subject to biases acting in opposite directions. With penalized regression, estimates are both shrunk towards the null of zero and, when averaging only the estimates for those betas that were selected to be included in the model, the averages are inflated due to the winner's curse. Further the effect of the winner's curse depends on the sample size as well as the underlying effect size. That is, the displayed values are the averaged estimated betas, computed among the ones that are actually found. The relative impact of these conflicting forces depends on the penalization method, with lasso estimates shrunk far more than IHT estimates. The nature of the distribution also influences the balance, with logistic regression most affected by the winner's curse in our examples. In order to better illustrate the complex interplay of the winner's curse and shrinkage, we expanded table 3 to include lasso regression estimates. We also added additional text to guide readers through these issues. Note that in our examples, IHT estimates are reasonable for effect sizes as small as 0.05 when the distribution is normal, Poisson or negative binomial and for effect sizes as small as 0.10 for logistic regression. These results are not surprising, when sample sizes are small, small effect sizes make predictors almost indistinguishable from random noise. The winner's curse operates in this regime and cannot be eliminated by IHT and any other selection method. The lasso results might appear to be less subject to the winner's curse, however, as we and others have found previously. The lasso estimates are severely shrunk for even for smallest values of beta (e.g. 0.03 with the normal distribution) and never provide reasonable coverage of the true values

4. Another important set of simulation they did is the simulation in the presence of LD between SNPs. However, related to my comment #1, this set of simulation is also not realistic. In particular, the settings that only allow LD within each block, constant number of causal variants within each block of non-zero effect, and constant causal effect size, strongly favour their model and are impossible to be true in practice. I suggest to do simulation based on the real genotype data (e.g. NFBC1966) to expose their method to the challenge of real LD structure in the genome.

Following the reviewer's criticism, we modified the doubly-sparse projection to handle different within-group sparsity levels, and hence changed the simulation routine so that there are variable numbers of causal SNP within each block. We updated Table 4 under this setting.

Furthermore, we adopted the reviewer's suggestion by using the first 30,000 SNPs of the NFBC1966 data as simulation target. We simulated 2 groups of size 2000, 5 groups of size 500, and 10 groups of size 100, representing genes of different length. The remaining SNPs are lumped into a final group representing non-coding regions. In all there are 18 groups. We randomly selected 1 large group, 2 medium group, 3 small groups to contain 5, 3, and 2 causal SNPs, respectively. Non-coding region also contains 2 causal SNPs. In all there are 19 causal SNPs. Effect sizes are randomly chosen to be -0.2 or 0.2. We ran 100 independent simulation studies under this setup with variable group sizes, where the large, medium, small, and non-coding groups are each allowed 5, 3, 2, and 2 active SNPs. The results are displayed below, and are available as Table 5 in the article.

	Ungrouped IHT		Grouped IHT	
	TP	FP	TP	FP
Normal	17.02	1.98	16.95	2.05
Logistic	15.67	3.33	15.8	3.2
Poisson	17.09	1.91	16.99	2.01
Neg Bin	17.19	1.81	16.95	2.05

The ungrouped IHT and grouped IHT perform essentially the same with these randomly constructed groups. Because we randomly assigned groups to the NFBC dataset, we are enforcing arbitrary boundaries, and hence arbitrary sparsity levels within groups. We agree with the reviewer that our results in Table 4 may be an ideal scenario since it enjoys perfect group information. Fortunately, the table above shows that, even in the worse case scenario where group information is completely ill-defined, grouped IHT does essentially no harm.

5. Tables 4 and 5 show the power of detection. How about the false positive rates in these cases?

We have updated Tables 4 and 5 to show both true positives and false positives.

6. Since they aim to develop a method scalable to large GWAS and such data sets are available in the public domain, the author should consider to choose a data set of large sample size (e.g. UK Biobank data) to convince readers about the superiority of their method. The data set they used, NGBC1966, with 5.4k samples and 360k SNPs is too small and outdated as compared to the current scale of GWAS, typically of 100k+ samples and millions of common SNPs.

In the revised article, we added a data analysis example using UK Biobank data. This example includes ~200,000 samples and ~500,000 SNPs with both quantitative blood pressure measures as well as dichotomous hypertension phenotypes. In the article we apply two regression models for the quantitative and binary hypertension phenotypes.

7. The authors indicate that their doubly sparse projection is instrumental in detecting causal variants in LD. If it works well, this could be very useful in fine-mapping. Can MeldelIHT be applied to fine-mapping? What's the accuracy of correctly identifying the causal variants in the presence of LD between them? How does it compare to the other fine-mapping methods, e.g. COJO (Yang et al 2012 Nat Gen)?

Based on our preliminary results with LD, we believe that the doubly sparse projection in MendelIHT along with appropriately chosen weights based on functional annotation could reduce the multicollinearity issues that plague traditional approaches to regression and help discern which SNPs in a group of SNPs are most likely to be truly associated (causal). However more work needs to be done before we can make such a statement in the current article and so we don't mention or refer to the possibility of using MendelIHT for fine mapping. It would be difficult to directly compare our approach to modern methods of statistical fine-mapping like COJO because much of this current fine-mapping work is based on using summary statistics from one or more GWAS (marginal SNP p-values or z-scores), making the assumptions that (1) any multi-collinearity among SNPs is entirely due to LD, and (2) there is a single causal SNP per region. Furthermore, they use a Bayesian approach with priors for the probability that a particular SNP is causal. Our methods require having the raw data, don't assume that SNP correlation is due to LD, allow for more than one causal SNP per group and are frequentist in nature.

8. They suggest to weigh the rare variants by the inverse of square root of heterozygosity, which implicitly assume a model of strong natural selection, that is, rare variants tend to have large effect size. Although there is increasing evidence of widespread natural selection on human complex traits, the relationship between SNP effect size and sqrt(heterozygosity) is not that strong (weaker than the linear relationship of -1) (Zeng et al 2018 Nat Gen; Schoech et al 2019 Nat Com). I concern the strong negative relationship suggested by the authors will overweigh the rare variants and introduce inflation in rare variant discovery. The validity and utility of this prior weighting strategy based on allele frequency need to be justified by simulations with realistic settings.

We thank the reviewer for noting that our phrasing was misleading. The choice of sqrt(heterozygosity) is meant to be illustrative, alerting readers to the ability to use weights if desired with our IHT package, rather than normative. We agree that those weights capture strong negative selection as we noted in our previous research [Zhou et al Pacific Symposium on Biocomputing]. In our software, we leave up to users to choose the weights they desire. These choices can and probably should be informed by prior biological knowledge. It's simply beyond the scope of our article to enter into a detailed discussion of how to choose informative weights. As the reviewer implies in their comments, the appropriate choice of weights will vary by the underlying genetic architecture of the trait.

9. In the real data analysis, they said they don't have binary traits so they threshold LDL as a binary trait. I wonder if they do the same classification to HDL, what do they find? Are the results consistent with quantitative HDL analysis?

We include the result of this analysis in the updated article where we dichotomize HDL into low and high using 60 mg/dl as the cut point. We were able to recover 4 SNPs that was identified in our continuous model. An additional association was found with rs9898058, but it is only known be to associated to an unrelated phenotype. The results are provided in the following table for the reviewer to inspect.

SNP	Position	Estimated beta
rs6917603	119293	0.20
rs9261224	119288	-0.06
rs9261256	119294	-0.06
rs3764261	275336	-0.06
rs9898058	284637	0.04

However, because we now include the UKbiobank results, we only briefly touch upon these results. Interested readers are referred to our GitHub (https://github.com/OpenMendel/MendelIHT.jl/blob/master/figures/stampeed/HDL/analyze_stampeed_HDL_binary.ipynb), which contains the full analysis and result.

10. It's not straightforward to me why IHT gives higher power and lower false positive rate than Lasso. It would be helpful if the authors can explain/discuss it in Discussion.

We thank the reviewer for letting us know we didn't provide enough detail. In the revised article, we highlight the reasons in the introduction instead of the discussion. In short, IHT offers better performance because it provides unbiased parameter estimates, while the lasso is a shrinkage operator that biases the estimates towards zero, leaving a lot of unexplained variance. False positives fill the unexplained variance gap.

Minor:

11. Table 2. Are the numbers percentages? If so, how can some of them be greater than 100? Why NAs for Lasso with negative binomial model?

The numbers are the total number of true/false positives. False positives can become very large (>100) when selected predictors are spurious associations. The glmnet implementation, which we used for other lasso models, does not accommodate negative binomial models. That is why the displayed values are NA. These explanations have been included in the revised article.

12. Briefly describe what zero-inflated Poisson regression is?

This suggestion has been incorporated into the revised article.

13. In introduction, the authors outline some drawbacks of lasso: "First, the l1 penalty tends to shrink parameters toward 0, sometimes severely so. Second, lambda must be tuned to achieve a given model size. Third, lambda is chosen by cross-validation, a costly procedure". However, these are also the facts for their IHT method (l0 seems to *introduce more shrinkage than l1 and their method also need cross-validation to estimate lambda). Do the authors also consider these as limitations of their method?*

It is true that IHT also requires cross validation, so IHT has no edge over lasso in this respect. Most of our parallel computing effort has focused on improving this aspect of IHT. However, the L0 penalty of IHT introduces *less* shrinkage than the L1 lasso penalty. This can be seen in Figure 1, where the solid diagonal line are the true beta values and the dashed lines as estimated beta values. The reader can now also compare directly the IHT and lasso estimates in Table 3.

14. "our code can handle datasets with 10^5 subjects and half a million SNPs". Modern GWAS are often empowered by millions of common SNPs through imputation and hundreds of thousands of individuals due to biobank efforts. Can their package handle such data sets?

In the updated article, we demonstrate that IHT can indeed handle biobank-scale datasets. We use IHT to analyze hypertension related phenotypes on UK Biobank data.

15. I am not familiar with the debiasing step, so it reads a bit obscure to me on this part. I hope the authors can explain more about what the problem is here and how this approach can solve the problem.

We thank the reviewer for noting that the discussion on debiasing is confusing. In the revised article, we deemphasized debiasing and simply mentioned it as an optional step one can take when we employ it. We have also included a few references for the interested readers.

16. Fig2 shows the computing time for different sample sizes. I wonder what the relationship is between computing time and the number of SNPs (and the number of causal variants).

The relationship is linear as well since the computational bottleneck is based on full matrix-vector multiplication which scales as O(np). This information is now in the revised article.

Reviewer 2

1. You mentioned in the introduction that multivariate approach offer better outlier detection and better prediction. It is not clear how it can help identify outlier. Can you add a reference or explain more detail? I agree that multivariate method is better in prediction. However, the prediction performance is not evaluated in the manuscript. Prediction error was mention in page 4 for figure 3, but the deviance residual from the fitted model is not a prediction error. To evaluate prediction, you need to split the data into training and testing. Since MendelIHT used iterative hard thresholding instead of lasso. The gain in computation time might comprise the prediction performance. IHT seek steepest descent, which may prone to overfitting. It is important to compare the prediction performacne of your method with LASSO or the method in refence [33].

We include a new reference for why multiple regression offers better outlier detection. As for the reviewer's second concern, we provide deviance residuals in figure 3. These provide one form of prediction error on the testing set. In this cross validation study, we split the data into training and testing sets. We have updated the text to describe this more clearly in the revised article.

2. Regarding on double sparsity, Lasso based approach would not limit the number of nonzeros in each group. MendelIHT can only pick a fixed number k nonzero coefficients in each group. I view this as an disadvantage as size of the group tends to vary and in reality it is not realistic to assume each pathway would contribute equal number of important predctors. I would expect that Group IHT have better prediction in real data than MendelIHT, as it wouldn't artifically set samller coefficents to 0. More discussion on this issue would stengthen the manuscipt.

We thank the reviewer for pointing out this drawback compared to group lasso. Based on the reviewer's criticism, we have modified doubly sparse projection so that it can handle varying numbers of non-zero coefficients in each group. Thus, group lasso no longer has this edge over doubly-sparse group IHT and as we point out in our responses to reviewer 1's criticisms, lasso penalties lead to shrinkage and worst model selection.

When compared to group-IHT, doubly-sparse IHT can also set the level k for each group high enough so that all SNPs come into play within a group. Therefore, the performance of group-IHT is a lower bound for doubly sparse IHT after our modification. We thank the reviewer for providing this good insight, which has been incorporated into the revised article. Our method also has the advantage of model interpretability. For GWAS, groups can be genes or pathways; hence they can contain thousands of SNPs. Doubly sparse groups select genes/pathways as well as SNPs within the genes, where as group IHT can only select genes/pathways.

We thank both reviewers for the comments and hope that our responses and revisions have covered all of their concerns.